Engineering Workshops

Requirements Engineering

A Practical Approach to Modeling and Managing Requirements

Course Guide

Version 20.1 (Slides Version 21)

About SSQC

William J. Deibler II and Robert Bamford founded SSQC in 1990 to support organizations in the definition and implementation of Software, Hardware, and Systems Engineering Practices, Software Quality Assurance and Testing, Business Process Reengineering, ISO 9000 Registration, and CMM/CMMI implementation. Their clients have successfully achieved ISO registration and advanced CMM and CMMI maturity levels.

Bob and Bill have developed and published numerous training courses, assessment and auditing tools, research papers, and articles on interpreting and applying the ISO 9000 standards and guidelines and the SEI Capability Maturity Model for Software. They were the principal authors and editors of **A Guide to Software Quality System Registration under ISO 9001**, and have served as active United States TAG members in the ISO/IEC JTC1 SC7 - Software Engineering Standards subcommittee, which is responsible for the development and maintenance of ISO 12207 and ISO 15504 (SPICE). Their latest book, **ISO 9001:2000 for Software and Systems Providers, and Engineering Approach,** was published in 2003 by CRC Press. The book joins their extensive portfolio of articles, which have appeared in McGraw Hill's **Quality Systems Update**, IEEE **COMPUTER**, McGraw Hill's **ISO 9000 Handbook**, **CrossTALK**, and **Software Marketing Journal**.

Since 1990, they have presented research papers and tutorials at over 50 national and international conferences, including those sponsored by the American Society for Quality (ASQ), the ESPI Foundation (ESEPG), Pacific Northwest Software Quality (PNSQC), the Software Publishers Association (SPA), Software Technology Support Center (STSC), the Software Engineering Institute (SEI) and Software Research Inc. Their courses have been offered through universities and professional associations, including the ASQ, the CSU Long Beach Software Engineering Forum for Training, Semiconductor Equipment and Materials International (SEMI), the Software Engineering Institute (SEI), UC Berkeley, and UC Santa Cruz.

William J. Deibler II has an MSc. in Computer Science and over 25 years experience in the computer industry, primarily in the areas of software and systems development, software testing, and software quality assurance. Bill has extensive experience in managing and implementing CMM-, CMMI-, and ISO 9001-based process improvement in software, hardware, and systems engineering environments. Bill is an SEI-authorized SCAMPI Lead Appraiser for CMMI.

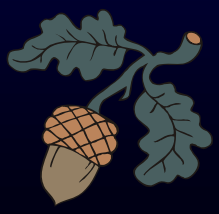
Robert Bamford has an MAT in Mathematics. In a professional career spanning more than 30 years, he has taught secondary and university Mathematics, and has worked in and managed training development, technical publications, professional services, and third-party software development. His experience also includes implementing a Crosby-based Total Quality Management System, implementing CMM-, CMMI-, and ISO 9000-based systems, and developing and facilitating workshops and courses.

Software Systems Quality Consulting
2269 Sunny Vista Drive, San Jose CA USA
+1-408-985-4476 http://www.ssqc.com

Engineering
Workshop Series

Requirements Engineering

A Practical Approach to Modeling and Managing Requirements

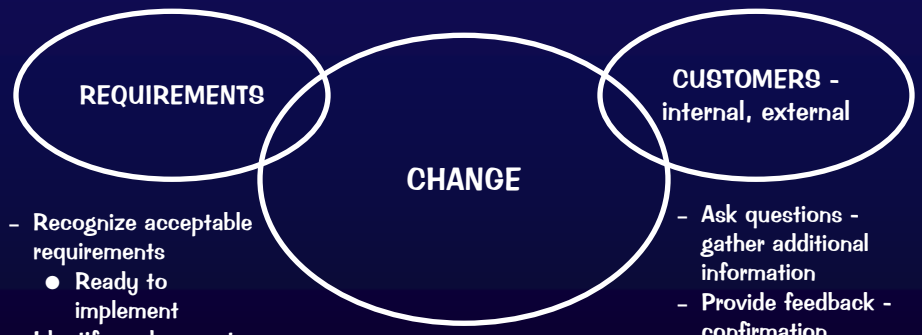


Objectives

- Prepare you to systematically
  - Mitigate current requirements problems
  - Prevent future requirements problems
  - Maximize efficiency in dealing with requirements and requirements changes
- Individual - immediate application
- Organization
  - Evolve infrastructure/processes
  - Identify obstacles, opportunities

2

Required skills - deal with:



- Recognize acceptable requirements
  - Ready to implement
- Identify and correct specific problems
  - Ask the right questions
  - Appear professional
- Identify and trace

- Requirements will change
- Customers will change
- No unnecessary changes
- Manage change - assess impact, prioritize, plan

- Ask questions - gather additional information
- Provide feedback - confirmation
- Elicit information - joint development of requirements

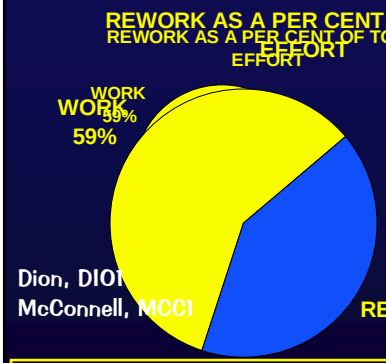
3 SS QC

Agenda

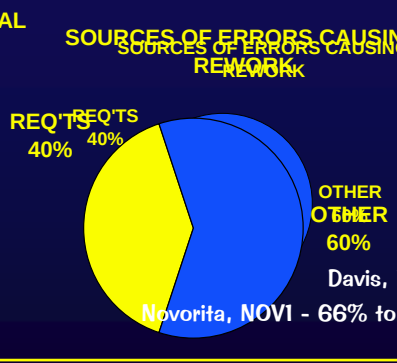
- Introduction
- The benefits of requirements engineering
- The impact of requirements
- Issues, questions, and priorities
- Requirements engineering
  - Processes
  - Techniques
  - Tools

4 SS QC

By the numbers: the impact of requirements



Dion, DIOT
McConnell, MCC1



Davis, DAV1,
Novorita, NOV1 - 66% to 55%

TYPES OF REQ'TS ERRORS

Incorrect fact	49%	
Omission	31%	
Inconsistency	13%	
Ambiguity	5%	
Misplaced	2%	Hooks, HOO3

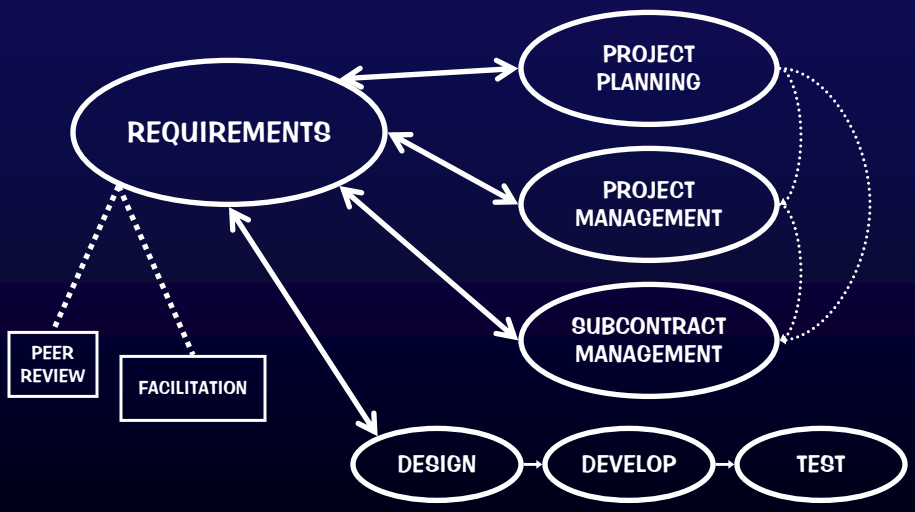
To fix requirements errors after deployment:
50x to 200x cost factor McConnell, MCC1

55% or more of the ... failures discovered by end users and system testers are caused by problems with requirements. The most probable causes are:

- Ambiguous words and phrases
- Incomplete statements
- Inconsistent functions
- Untestable functions
- Untraceable functions
- Undesirable design impositions

Robert M. Poston, **Generating Test Cases from Use Cases Automatically**

The impact of requirements



6 SSQC

Team Exercise:

Issues, questions, and priorities

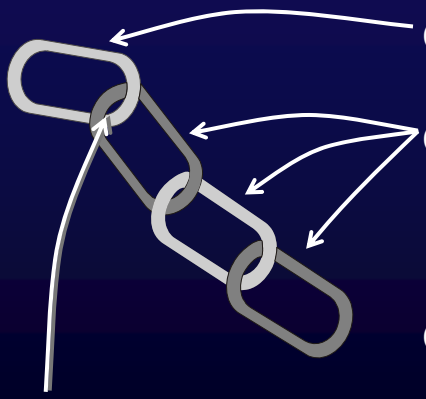
- Related to requirements - wherever they come from
  - Customers
  - Systems Engineering
  - Marketing
- Top five issues and questions - in priority order. Base priority on:
  - What issues offer the greatest opportunity for improvement?
  - Recurrent issues

- Confusion about what a requirement is
- Inadequate customer involvement
- Vague and ambiguous requirements
- Unprioritized requirements
- Building functionality no one uses
- Analysis paralysis
- Scope creep
- Inadequate requirements change process
- Insufficient change impact analysis
- Inadequate requirements version control

© Software Systems Quality Consulting
2269 Sunny Vista Drive, San Jose CA 95128
All rights reserved.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Defining “requirement”



Confirmation, agreement and commitment

(1) A condition or capability needed by a user to solve a problem or achieve an objective

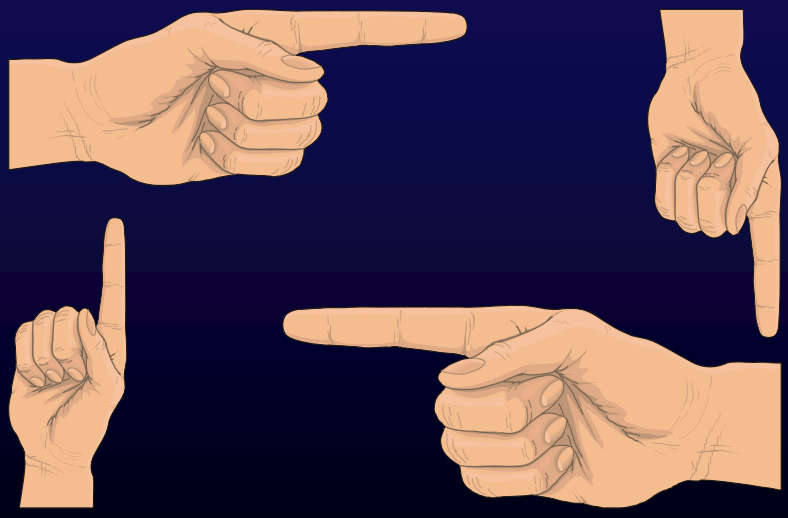
(2) A condition or capability that must be possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed document *[from ...]*

(3) A documented representation of a condition or capability as in (1) or (2)

IEEE 610.12-1990 **Standard Glossary**

THE HISTORY OF REQUIREMENTS MANAGEMENT

Common themes - responsibility



© Software Systems Quality Consulting
2269 Sunny Vista Drive, San Jose CA 95128
All rights reserved.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

THE HISTORY OF REQUIREMENTS MANAGEMENT

Uncommon theme

TO SOLVE THIS PROBLEM

Strategy and philosophy

❷ Educate our customers as to what we need and why we need it
- Benefits to them
- Provide tools and training

❶ Get our own house in order so that
- There is a benefit to our customers and to us
  - If we don't listen, why should they bother telling us what they want
- We know what we need

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Team Exercise

Inside every complaint is a need.

Test Drive: User-originated input

SHR INCORPORATED

Formal Input Tracking System **FITS**

		FITS Identifier **FID**	1204-006-030422.183-0454Z
	UNIVERSAL PERSONNEL CODE **UPC**	**NAME**	**LOCATION**
ORIGINATOR	015477408	Skilling, P. W.	SM2-06 DEPT cc

	PRODUCT IDENTIFIER PID	**VERSION**
SYSTEM INFORMATION	Operating Budget Target Tracking Application - OBTTA	5.01.0004

PROBLEM OR REQUEST X Problem Request	The Extended Money Values for approved Inter-Depot Transfer Transactions are not permanently reflected in the on the Depot Working Budget Report. I reported this problem t/r #552950 which was cancelled w/o explanation. This is critical since depot managers use the budget report when making approval decisions.
IMPACT	Depot managers may approve requests when budget not there.

	CRITICAL	**IMPORTANT**	**ROUTINE**
ORIGINATOR PRIORITY		X	

13 SSQC

Team Exercise

Background: FITS

SHR INCORPORATED

Formal Input Tracking System **FITS**

		FITS Identifier **FID**	1204-006-030422.183-0454Z
	UNIVERSAL PERSONNEL CODE **UPC**	**NAME**	**LOCATION**
ORIGINATOR	015477408	Skilling, P. W.	SM2-06 DEPT cc

	PID	**VERSION**
SYSTEM INFORMATION	Op king Application -	5.01.0004

PROBLEM OR REQUEST X Problem Request	Th r approved Inter-Depot Transfer Transa ns are not permanently reflected in the on the Depot Wo king Bu s problem t/r #552950 which was cance s critical since depot managers king approval decisions.
IMPACT	Depot mana en budget not there.

	CRITICAL	**IMPORTANT**	**ROUTINE**
ORIGINATOR PRIORITY		X	

PROBLEM OR REQUEST
Select one
- Problem = System not working correctly
- Request = New capability needed; improvement

IMPACT
How does this problem affect your (unit's) performance?
How will this change affect your (unit's) performance?

PRIORITY
- Critical = Impacts current mission readiness, compliance with regulation
- Important = Improves efficiency, reduces error rates
- Routine = Request consideration

14 SSQC

Team Exercise

Background - OPERATING BUDGET TARGET TRACKING APPLICATION (OBBTA)

You download his data and determine that Skilling has the correct, current, version.

You check FITS and determine that:

- TR 552950 was cancelled; the reason was “system operating as designed”. The SHR contact is not available.
- You can’t find another problem report related to the problem described.

You walk down the hallway to the Standard Systems Simulation Lab (S^3L), load version 5.01.004 of OBBTA and Skilling’s data, and determine that:

- Sample inter-depot transfers show up correctly in the Depot Working Budget Report (DWBR-0021).

Finally, because you are thorough, you accelerate time by a factor of 1,000 to simulate running the system for a month and determine that:

- Sample inter-depot transfers still show up correctly in DWBR-0021.
- Sample inter-depot transfers show up correctly in the Regional Monthly Roll-up Report (REG-0033).

YOU HAVE (AT LEAST) 2 CHOICES

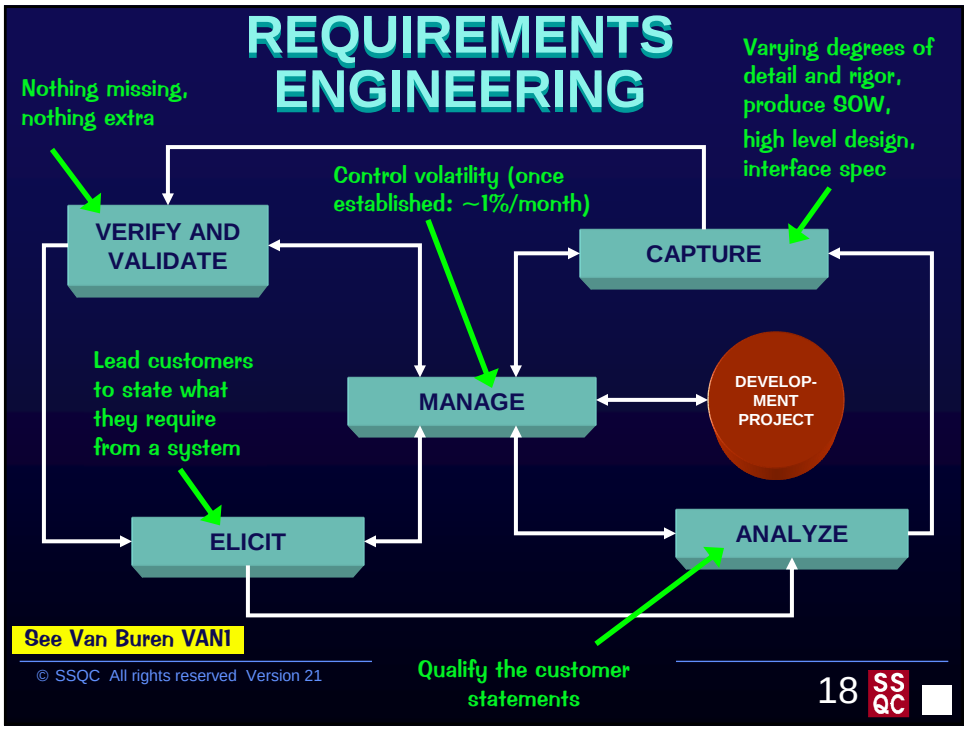
OR ... DO YOU FEEL LUCKY TODAY?

15 SS QC

16 SS QC

Defining a requirements engineering process

- Requirements engineering
- Requirements identification



Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

ELICIT

- RFP
- Interviews
- Focus groups
- Questionnaires, surveys
- Brainstorming
- Role play
- Review incident reports, enhancement requests
- Joint authorship
- Benchmark - similar or competing systems
- Prototype
  - Throw-away: when critical features or architecture not well understood
  - Evolutionary: to refine, to understand non-critical features

Davis DAV1, SEI1

CAPTURE, VERIFY, VALIDATE

- **Text**
  - Incident reports
  - Enhancement requests
  - Documents (System Requirements Specification, Solicitation and Proposal, Statement of Work, Test Plan, Test Cases, etc.) - electronic or paper
- **Prototype**
- **Graphical representations**
  - Process maps
  - Test cases
- **Database**
- **Provide**
  - Verification and validation
    - Review - upstream (confirmation) and downstream (sufficiency)
    - Test - at various stages; prior to deployment
  - Consistent communication - across time and location
  - Baseline - for plans and for content

19 SSQC

Identify requirements: attributes and data

- Enumeration
- Type
- Necessity
- Stability
- Allocation
- Parent (requirement)
- Source (document, paragraph)
- Rationale
- Verification (method, documents)
- Changes
- Current status
- Planned release

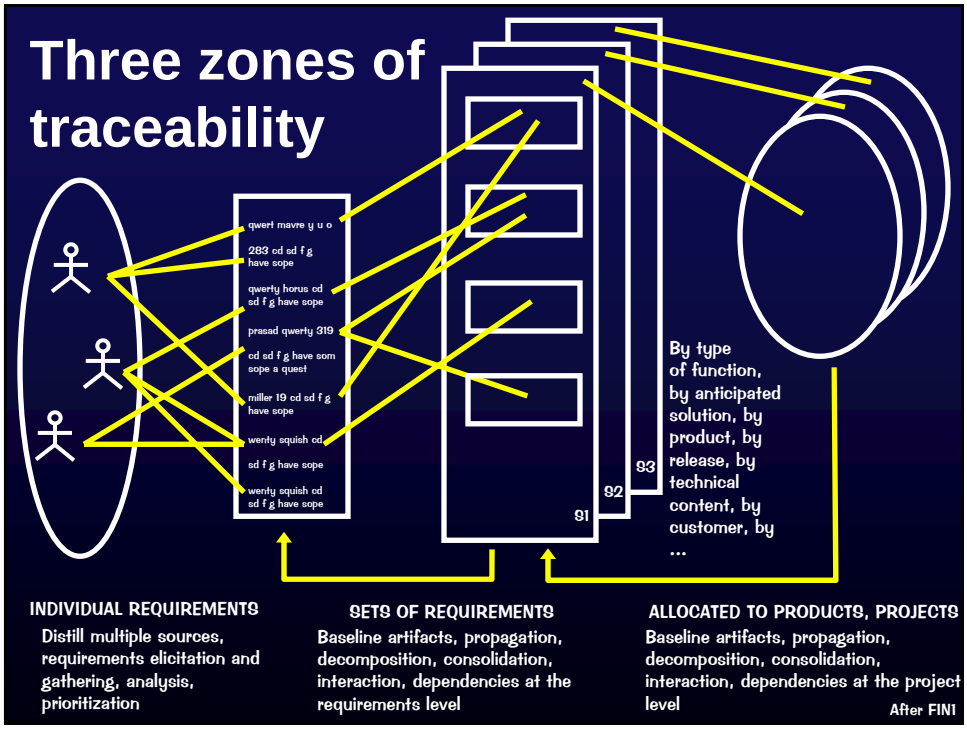
- Program (process, legal, schedule)
- Technical
  - Functional (user, mode)
  - Exception handling (user, mode)
  - Performance (quantitative)
  - Design constraint (environment)
  - Interface (interact with user, external components and other internal components)
  - Communications (connectivity, access)
  - Computer security and access
  - Backup and recovery
  - Implementation (conversion, installation, hardware acquisition)
  - Information (examples, other)

Essential, Desirable, Optional

High, Medium, Low (Risk of change)

To be defined
To be reviewed
Defined
Approved
Planned
Verified
Deleted
Deferred

Kar KAR1, DOE1, IEEE 830



Robust identification

- Decompose, merge requirements
- Trace allocation from source through
  - Projects
  - Components
  - Products

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

The allocation conundrum

BSC offers three products: a word processor, a spreadsheet, and a presentation tool. Two of the products are from separate companies that were acquired by BSC; the third was developed by BSC. The three products all run under Windows QT and are written in C and C++, but otherwise, they have virtually nothing in common. They are sold as a single suite. The products are maintained by three separate development teams, which compete for budget based on revenues from the sales of the suite (for which they all try to take full credit).

Market research has proven that customers for the word processor and the presentation tool urgently need the ability to draw flow charts with live connectors (e.g., they are "attached" to shapes so that they move when the shape is moved). The competition has such a capability.

1. Given just the above information, brainstorm at least four theoretically-feasible strategies (good, bad, or otherwise) for allocating the requirement to the engineering teams. Include any additional constraints (e.g., design).
2. For the two most likely strategies, state the potential risks and benefits to BSC.

Defining sets of requirements

- The Requirements Specification
  - Establish the basis for agreement between the "customer" and the development organization on what the delivered product is to do
  - Reduce the development effort
  - Provide a basis for estimating costs and schedules
  - Provide a baseline for validation and verification
  - Facilitate transfer
  - Serve as a basis for enhancement and change
- Engineering establishes control of "customer" input
  - Elicit (additional) information and clarification
  - Lead the customer - collaborate
  - Confirm understanding and agreement - active listening

Defining sets of requirements (cont.)

- Amount of work depends on
  - Source
    - Technically sophisticated “customer”
    - Someone with an idea in search of support
  - Received content
    - Back of a napkin (create, collaborate)
    - Detailed Statement of Work - SOW (verify)

Guidance on processes and documents

Systems

- IEEE Std. 1233-1998 **IEEE Guide for Developing System Requirements Specifications** [IEE3]
- ISO/IEC 15288 **System Engineering - System Life Cycle Processes** (final draft) [ISO1]
- EIA/IS 731 **Systems Engineering Capability Model** [INC2]
- CMU/SEI-2002-TR002 **Capability Maturity Model Integrated (CMMI) for Systems Engineering/ Software Engineering, Version 1.1** [SEI3]

Software

- ISO/IEC 12207 **Software Life Cycle Processes** (parts 1, 2, and 3 in IEEE/EIA 12207) [IEE4]
- IEEE 830-1998 **IEEE Recommended Practice for Software Requirements Specifications** [IEE1]

GENERAL SOURCES

- Cooperative Systems Engineering Group (CSEG), Lancaster University [LAN1]
- International Council on Systems Engineering (INCOSE) [INC1]

- Trigger a controlled explosion of creativity
  - No unnecessary constraints
  - Not a data dump
  - Clearly organized and presented
    - TBDs – sometimes necessary; clearly identified
    - Assumptions – clearly stated
    - Redundancy and repetition - eliminated
    - Focus – priorities identified
    - Technical and non-technical requirements

Outline of a generic SRS

1 Introduction
- 1.1 System purpose
- 1.2 System scope
- 1.3 Definitions, acronyms, and abbreviations
- 1.4 References
- 1.5 System overview

2 General System Description
- 2.1 System context
- 2.2 System modes and states
- 2.3 Major system capabilities
- 2.4 Major system conditions
- 2.5 Major system constraints
- 2.6 User characteristics
- 2.7 Assumptions and dependencies
- 2.8 Operational scenarios

3 System capabilities, conditions, and constraints

4 System interfaces

System Requirements Specification

IEEE 1233-1998

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Outline of a generic SRS

1. Introduction
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 Definitions, acronyms, and abbreviations
   - 1.4 References
   - 1.5 Overview
2. Overall description
   - 2.1 Product perspective
   - 2.2 Product functions
   - 2.3 User characteristics
   - 2.4 Constraints
   - 2.5 Assumptions and dependencies
3. Specific requirements

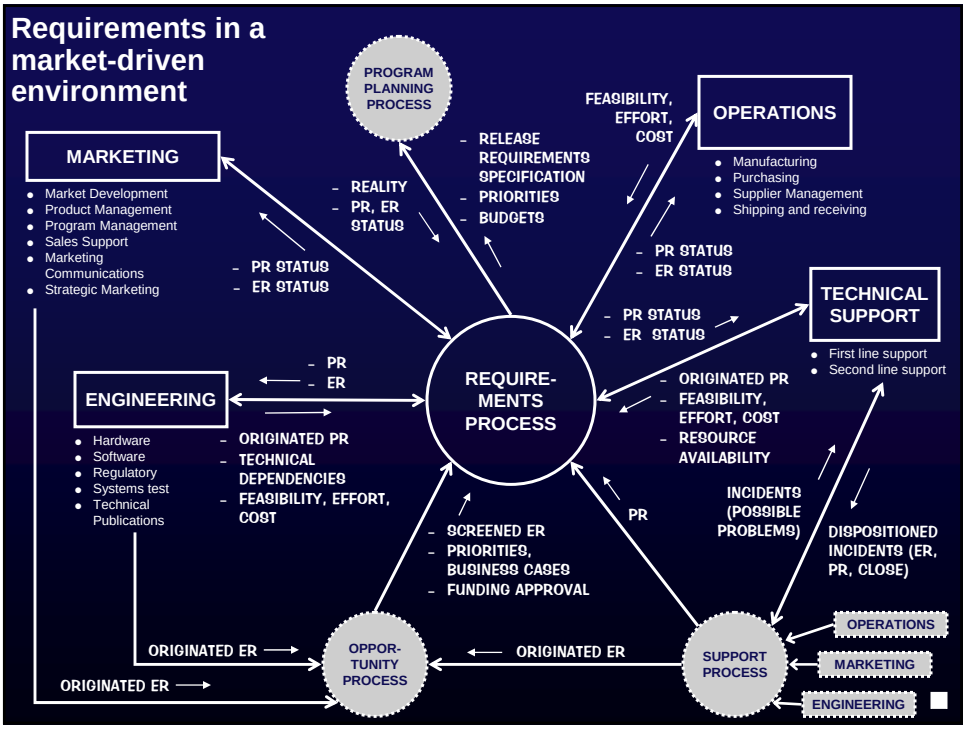
Appendixes

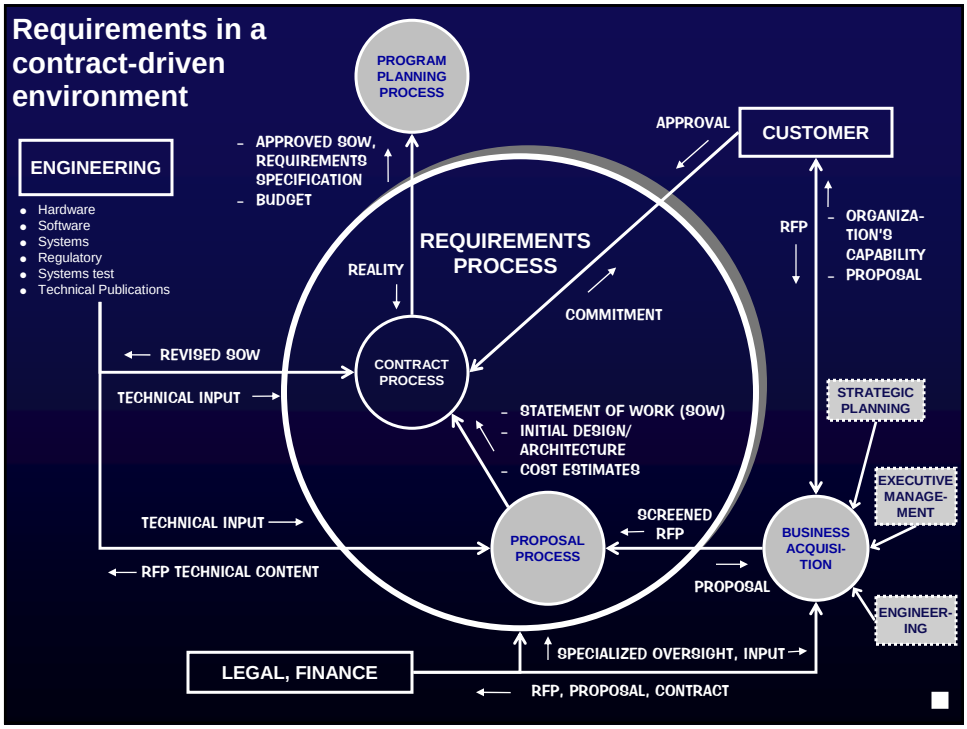
Index

Software Requirements Specification

IEEE 830, 1998/IEC 12207

Requirements in a market-driven environment





Elements of a requirements process

- Baseline - set
- Version - individual and set
- Identification (enumeration) – individual and set
- Capture
- Visible - accessible
- Responsibility and authority - approval and change (Change Control Board [CCB])
- Internal peer review; external review
- Address all types of requirements – technical and non-technical
- Requirements specification - package of requirements, interactions, dependencies, etc.

32 SSQC

TEAM EXERCISE
How's YOUR process?

- Describe how your current requirements process(es) address the process attributes
  - 5 minute presentation
  - Consider key activities - inside and outside engineering
  - Evaluate anything that's missing (e.g., unnecessary because …)

PROCESS ELEMENTS

- ✔ Baseline
- ✔ Version
- ✔ Identification
- ✔ Capture
- ✔ Visible
- ✔ Responsibility and authority
- ✔ Review - internal, external
- ✔ All types
- ✔ Requirements specification

What are the attributes of a requirement?

	Individual	Set
✔ Clear (unambiguous and complete)	X	X
✔ Correct	X	
✔ Consistent		X
✔ Singular	X	
✔ Testable, verifiable	X	
✔ Feasible	X	X
✔ Design independent	X	
✔ Traceable	X	

© Software Systems Quality Consulting
2269 Sunny Vista Drive, San Jose CA 95128
All rights reserved.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Clear: unambiguous and complete

- Unambiguous: representatives of all affected parties should be able to read the statement and come to a single, consistent interpretation
- Completely describes performance, functionality to be delivered - not just the undesirable behavior to be eliminated or replaced.
  - All types considered - including exceptions (set)
  - Internal references resolved (set)
  - No implied elements missing (set)
  - Contains the information needed for all affected parties to agree to begin work on the next stage in the life cycle

- ***Beware:*** "include, but not limited to", "support", “and/or”, “and”, “this”, “it”, “appropriate”, “consider”, “any”, complex grammatical construction

BEFORE
The report isn’t coming up quickly enough and it isn’t available where we need it.

DURING
- ✔ What is quickly enough? Where is it needed?
- ✔ Is it coming up where it shouldn’t?
- ✔ Are there security concerns?
- ✔ What form is it needed in?
- ✔ What if a printed form is requested, but no printer is available?

AFTER
The report shall be generated for viewing, printing, or both, when requested by an authorized user at any networked workstation. The report shall be available for on-screen viewing and search within 10 seconds; a complete printout shall be available within 45 seconds of request.

BEFORE
When the record is not completely updated, the data may be incorrect.

DURING
- ✔ When is a record “not completely updated”?
- ✔ What does incorrect mean?

AFTER
If the operator exits the system with F8 prior to completing a transaction, the data in the system may not match the data entered.

FURTHER AFTER
- ✔ Why exit?
- ✔ Do we need to keep partial records?

BEFORE
Sound an alarm for two minutes.

DURING
- ✔ And then?

AFTER
?

BEFORE	DURING	AFTER
■ Allow the operator to correct the date.	■ The operator? What entry device? Security and integrity implications? Logging? Why? What if the correction is wrong?	■ ?
■ The goal of this feature is to limit access to utilities so that a customer can purchase distinct service levels. Possible layers would be: "self service", "basic service", "full service", and "premium service".	■ What are "layers" versus "levels"? What should go in each layer? Layers are cumulative, aren't they? Who decides what goes in a layer - or a level?	■ ?

37

Correct

- Accurate (and complete) as determined by the customer and user or their representative(s)
  - Is it the right thing to do (priority and content)?
  - Does it have value for the customer?
- Examples of customer responses indicating incorrect requirements derived from customer input
  - Well, *technically* that is what we asked for, but … .
  - That's what we need right now.
  - Where did you see that?
  - That's a nice idea.
  - That is not open to discussion.
- Advise, facilitate, escalate

Correct?

- Modify CTRL+V to jump to the verify screen
- Change all screen colors to black
- This release will focus on both pediatric and adult imaging
- Lock out inter-depot transfers
- Don't waste time testing everything. We need it now.

38

Consistent

- Not in conflict with other requirements (technical or non-technical, policies, regulations, etc.).
  - Do not duplicate or overlap
- Examples
  - *Lock out inter-depot transfers.*

  and
  - *Cut turn-around time on inter-depot transfers.*
- Conflict is natural and inevitable
  - Advise
  - Broker
  - Facilitate
  - Escalate

Singular

- Does not contain multiple requirements. A non-complex requirement. Sufficient granularity to specify a single requirement.
  - "and", "or", commas, semicolons
- Examples:
  - The training is wrong and the system gives us the wrong report.
  - We need to be able to enter the transactions and see the audit report at any terminal.
  - Web-enable (which word don't you understand?)
- Analyze, decompose, factor

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Testable, verifiable

- Description is sufficient that it is possible to determine whether the statement is properly implemented in the product.
- *Ambiguous requirements may be untestable*
  - *Depending on the range of ambiguity.*
- *Incomplete requirements are untestable.*
- *It may be infeasible or impossible to test an unambiguous, complete requirement.*

- ***Beware:*** minimize, maximize, fast, user-friendly, easy, sufficient, relevant, acceptable, adequate, quick, industry standard, normal

BEFORE
The number of calls that cannot be initially completed should exceed the competition.

DURING
What are the current rates competitors achieve?
Wait a minute ...

BEFORE
Under Windows, the new system shall provide all of the functionality of the DOS-based system, with a user-friendly GUI and easy-to-use screen architecture.

DURING
User friendly?
Easy to use?
Conforms to Microsoft Corporation, **Microsoft Windows User Experience**, 09/08/1999, ISBN 0-7356-0566-1.
Prerequisites for users? Training?

BEFORE
Mode 2 imaging must be competitive with or superior to the IQ 55C and the HUFFY HighFive system.

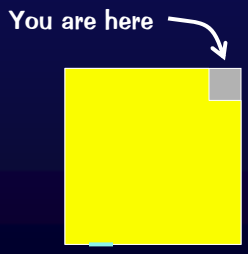
DURING
Specific functions? Performance criteria or targets? Tradeoffs? What are competitive futures?

Feasible

- It is possible to implement the requirement and/or set within the known capabilities and limitations of the system and its environment
  - Technically, and
  - Within costs and schedule constraints
- Feasible is based on what we know about the requirements as stated, not on whether we have all the information we need to move to the next phase in the life cycle
- The first phase of a program may be to gather additional detailed information

Design independent

- Describes “what”, not “how”
  - User need or problem - not a solution
  - Function not operation
- Ask *WHY?*
- Accepted exceptions are constraints
- Exploring the constraint frequently exposes additional requirements
  - *What I really need is …*
  - *Everyone is doing it.*
  - Incompletely considered implications
  - You gave me what I asked for, not what I need



Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

- Use Windows NT
- Do in 64MB of RAM
- Add a field for standard industry classification (SIC) to the employee's competency record
- Provide a common data base for Customer Relationship Management applications
- Control work stations with access cards.
- Web-enable the application
- Direct printer support will be available for all calculation packages with reporting features
- The system shall verify that the machine status is "OFF LINE" before allowing repair parts to be ordered

WHY?

Apply common sense in enforcing design independence

- Consider the source of the proposed requirement
- Consider the nature of the project - maintenance versus new development

A Dialogue (of sorts)

© Software Systems Quality Consulting
2269 Sunny Vista Drive, San Jose CA 95128
All rights reserved.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Traceable

- Able to link the requirement to a specific origin
- Beware
  - Adding this feature was recommended by Direct Digital Systems, Inc.
  - As required by Federal EEO regulations
  - This requirement is from the Direct Digital Systems Request for a Proposal (RFP) received last month.
  - Everybody thinks this is a problem.
  - The competition …
  - The market ...

You don't trust me?
I.M. Hurt

Evaluate a requirement to determine ...

- What improvement it may need
- What strategies to employ to improve it
  - Discuss
  - Prototype
  - Investigate
- A realistic level of confidence in estimates based on the requirement
- When it is sufficient to proceed
  - Can I accommodate the TBDs?
  - *I may need X or Y*

Evaluate a requirement to establish ...

- A common language to discuss requirements (types and attributes)
  - Writing effective requirements is not easy
- Realistic customer expectations of what we need

Making it better

If It's Not	Then we have to …
Clear, Test-able, Design Independ-ent	Search for the "real" requirement through interviews with the originator(s) or subject matter expert(s) (SME). If appropriate, assist the originators through research, models, demos, use cases, or prototypes. Validate the requirement with the originator or SME. Track down the missing information; query the originator.
Consistent	Resolve the conflict by determining relative priority or through negotiation with affected stakeholders. Change or eliminate conflicting or overlapping requirements.
Correct	Determine the "right" requirement. Validate the requirement with the originator/user or their representative(s). Escalate/expose for a decision/solution.
Traceable	Pinpoint the source. If appropriate, compare it to its source to determine if the stated requirement meets the intent of the source (originator, policy, regulation, law, etc.)
Feasible	Refer, defer, identify trade-offs (schedule, performance - What can we do?), decline, play chicken.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Techniques

- Evaluate textual statements to identify adequacy and opportunities for improvement
- Elicit, capture, and confirm complex requirements through graphical representation

A few suggested "do"s and "don't"s

DON'T

- Guess
- Impose or assume
- Rely on prior knowledge
- Live with it
- Jump to a solution
- Make a mountain out of a molehill

DO

- Discuss
- Propose and verify
- Draw on prior knowledge
- Present the risk of poor requirements and the benefits of clear requirements
- "Read" what's there – the facts
- Respond to customer priorities and needs
- Ensure you communicate clearly with your customer(s)
  - Internal and external
  - Not too much; not too little
  - Anticipate misunderstanding
  - Peer review before release

© Software Systems Quality Consulting
2269 Sunny Vista Drive, San Jose CA 95128
All rights reserved.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Test Drives - three parts

EVALUATE INITIAL INPUT,
IDENTIFY PROBLEMS,
PREPARE QUESTIONS

REVISE QUESTIONS,
PREPARE FOR AND
CONDUCT AN INTERVIEW

WRITE
REQUIREMENT(S)

Test drive 1

From a request submitted for Flak-Trak, an automated hot-line system. Flak-Trak is installed in 7x24 help-desk centers that handle customer telephone calls.

After 12 minutes we need FlakTrak to automatically escalate any incoming call that has not been answered to the hot-shift line lead.

© Software Systems Quality Consulting
2269 Sunny Vista Drive, San Jose CA 95128
All rights reserved.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Test drive 2

From a bug report for an inventory management system your group maintains. The report is listed as originated by M. Wagner at the Denver Depot.

When the Flash utility is used to add and item to the Depot Parts List (DPL), the item doesn't show up in a DPL Query. But when DPL Add is run, it says the item is already there. If an item is added with DPL Add everything works fine. Except Flash overwrites an item with the same part number.

Test drive 3

From an enhancement request for a data warehouse query application your group maintains. The enhancement request comes from the Southern Region Field Sales Support organization.

The system shall allow users at IBM-compatible PCs with 32 MB of memory to construct report templates without being connected to the network and the system shall work the same.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Test drive 4

From a problem report for a data warehouse query application your group maintains. The problem report comes from the Southern Region Field Sales Support organization.

On the On-Hand Inventory screen, OHI-24, PARTS AVAILABLE BY PART NUMBER, the field for “Total Units Available at all Sites” is not updated so information is not available to personnel checking inventory at one location.

Test drive 5

From the statement of work for a warehouse fire detection system.

The system shall poll up to 128 sensors divided in up to 4 zones, with a maximum of 28 sensors per zone, at a rate of once every 30 seconds, except under high-load conditions, in which case the polling may by zone may take up to 60 seconds, or 90 seconds, if necessary, but an exception record will be logged on a journal printer (at the operator console) if the length of the polling cycle exceeds the above stated rate for more than 3 minutes and 10 seconds. After more than 4 minutes of extended polling, an audible alarm sounds.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Test drive 6

From a proposal approved as a top priority by Product Marketing, to consolidate two systems your group maintains.

Develop SuperSystem, a single, integrated system for managing inventories at all repair depots and regional warehouses. Start with and use most of the Regional Warehouse Distribution Center System (DCS) on-hand inventory valuation and distribution capability. Incorporate repair-kit management functionality from the Depot Stock Management System (SMS). Develop real-time update and data replication functionality to replace current batch reporting under DCS and SMS. Develop interfaces to use bar code technology wherever applicable.

59

Test drive 7

From an e-mail from the CIO at a major customer, which has installed your comprehensive manufacturing plant management system at 14 sites world-wide.

All applications need guidance for Application Administrators to perform verification of database restore and procedures for recovery upon catastrophic system failure. Guidance should be in the form of a Manual that be referred to from the System Administrators Manual and contain proper restart, restore, and validation procedures to be used by Administrators after recovery of system prior to letting all users back on system.

60

Test drive 8

From the statement of work for which you are preparing a response.

Starting with and using the functionality of the San Corolla City System for tracking citations, develop a system which will consolidate all citations issued in Costanoa County by any City police department (in the County), or by the County sheriff. The citation should be automatically referred to the correct Municipal or County Court, which will be responsible for recording the disposition of the citation. Currently, data on unpaid fines is manually reported monthly by each Court Clerk to the State Department of Motor Vehicles so that drivers license renewals and vehicle registrations can be withheld until any outstanding items are resolved to the satisfaction of the Court with jurisdiction. The new system should automate this reporting.

61 SS QC

Test drive 9

From a problem report submitted by the day-shift supervisor at a customer site.

When I use the Web3M v6 to record a MACHINE DOWN problem with a piece of mountable equipment (e.g., sorter, stapler) which can be moved from one copying machine to another, the problem stays against the machine as well as the piece of mountable equipment. So I can't put the machine back on READY status so I can schedule work for it. Right now when I move the broken mountable equipment, I delete the original DOWN problem as a "misdiagnosis" and open a new problem against a fake machine I added to Web3M and which everyone knows to show as always DOWN. This really messus up our equipoment inventory and our maintenance numbers.

62 SS QC

Test drive 10 - Evaluate

From a change proposal received from P. Blanchard at the TAS Travel Office for TAS, a Travel Accounting System your group maintains:

When a travel request is processed from the "pending transactions" screen, then saved and closed, the next travel request is not highlighted.

A few suggestions for authors and reviewers of text-based requirements

- The correct choice of words and phrasing is important, but
  - Supply examples, discussion, pictures - identified as supplementary, goals
- Focus on the audience
  - What is necessary? Useful?
- Focus on **cause and effect**
  - Results - What?
  - Means - Who?
  - Methods - How? (function not operation)

© Software Systems Quality Consulting
2269 Sunny Vista Drive, San Jose CA 95128
All rights reserved.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Communication, not great literature

Be aware of ...

- Four words
  - It
  - This
  - That
  - Which

© Software Systems Quality Consulting
2269 Sunny Vista Drive, San Jose CA 95128
All rights reserved.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

For example

- A component of MARKER provides the necessary communications protocol to communicate with the 15504 Data Sensor on-board self-diagnostic unit. It is written in a combination of Assembly Language and C to run in DOS or in full-screen DOS mode.
- It was recommended by the Sales Order Admin Supervisor at Martin Mariposa Company to add the feature on Customer Order Entry Screen to show the last two transactions.

- Automated routing shall allow the maintenance technician to submit the requisition for approval to the shift supervisor and for this person to forward the approved request to the central stock room to be filled.
- A copy of the release form is attached to the distribution media, which is retained by the project software development manager or lead.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Be aware of ...

- Four words
  - It
  - This
  - That
  - Which
- Passive voice

- User accounts are created by generating a user list from Microsoft Outlook prior to 8 o'clock between 7:53 and 7:58 Mondays through Fridays

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Choice of words

- Standardize terminology
  - Within a set of requirements - across sets of requirements
    - Acronyms
    - Examples

- Make clear what is required and what is optional
- Avoid weasel words - ambiguous, no agreed-upon usage
  - May, should, recommend, request, can, try
  - Consider
  - Guideline
  - And/or, either, or
- Focus on what you want, not what you don't want
  - Do not treat incoming inventory records processed through Inventory Count Transfer as final count transactions (unless count quantities match recorded quantities).
- Grammar, spelling, and constriction
  - Maintain confidence
  - Avoid distraction

Recommended Usage

Requirements: shall
Facts: is, are, will
Goals: should

- Read it before you send it.
- Have someone else read it.
  - Apply Peer Review tools and techniques

Communicating with customers

- Gaining support for requirements elicitation
  - Establishing expectation and ground rules
- Eliciting requirements

© Software Systems Quality Consulting
2269 Sunny Vista Drive, San Jose CA 95128
All rights reserved.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

The data around the increased quality and reduced costs of using JAD-like workshops are impressive.

According to Capers Jones, … 60% of defects originate in the requirements and design phases, early facilitated workshops reduce those defects by 20% to 60% … reduce the risk of scope creep from 80% to 10% … provide 5% to 15% overall time savings.

Ellen Gottesdiener, *Decoding Business Needs*, **Software Development**, Vol. 7, No.12, December 1999, page 28; quoting
Capers Jones, **Assessment and Control of Software Risk** (Prentice Hall, 1994)

The impact of requirements (.25)

16 Critical Software Practices for Performance-Based Management

Project Integrity

1. Adopt continuous risk management.
2. Estimate cost and schedule empirically.
3. Use metrics to manage.
4. Track earned value.
5. Track defects against quality targets.
6. Treat people as the most important resource.

Product stability and integrity

7. Inspect requirements and design.
8. Manage testing as a continuous process.
9. Compile and smoke test frequently.

Construction Integrity

10. Adopt life cycle configuration management.
11. Manage and trace requirements.
12. Use system-based software design.
13. Ensure data and database interoperability.
14. Define and control interfaces.
15. Design twice, code once.
16. Assess reuse risks and costs.

[EVA1] Michael Evans

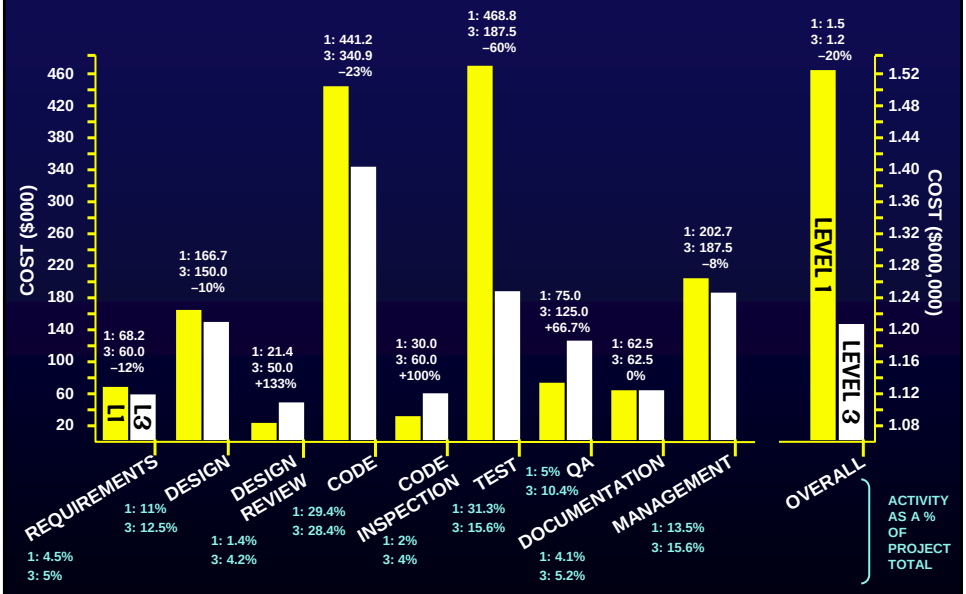
The impact of requirements (.600)

	RISK ITEM	RISK MITIGATION TECHNIQUES
1	PERSONNEL SHORTFALLS	STAFFING WITH TOP TALENT, JOB MATCHING; TEAMBUILDING; CROSS-TRAINING; PRE-SCHEDULING KEY PEOPLE; MORALE BUILDING
2	UNREALISTIC SCHEDULES AND BUDGETS	DETAILED, MULTISOURCE COST AND SCHEDULE ESTIMATION; DESIGN TO COST; INCREMENTAL DEVELOPMENT; REUSE; REQUIREMENTS SCRUBBING
3	DEVELOPING THE WRONG FUNCTIONS	ORGANIZATION ANALYSIS; MISSION ANALYSIS; OPS-CONCEPT FORMULATION; USER SURVEYS; PROTOTYPING; EARLY USERS' MANUALS
4	DEVELOPING THE WRONG USER INTERFACE	TASK ANALYSIS; PROTOTYPING; SCENARIOS; USER CHARACTERIZATION (FUNCTIONALITY, STYLE, WORKLOAD)
5	GOLD PLATING	REQUIREMENTS SCRUBBING; PROTOTYPING; COST-BENEFIT ANALYSIS; DESIGN TO COST
6	CONTINUING STREAM OF REQUIREMENT CHANGES	HIGH CHANGE THRESHOLD; INFORMATION HIDING; INCREMENTAL DEVELOPMENT (DEFER CHANGES TO LATER INCREMENTS)
7	SHORTFALLS IN EXTERNALLY FURNISHED COMPONENTS	BENCHMARKING; INSPECTIONS; REFERENCE CHECKING; COMPATIBILITY ANALYSIS
8	SHORTFALLS IN EXTERNALLY PERFORMED TASKS	REFERENCE CHECKING; PRE-AWARD AUDITS; AWARD-FEE CONTRACTS; COMPETITIVE DESIGN OR PROTOTYPING; TEAMBUILDING
9	REAL-TIME PERFORMANCE SHORTFALLS	SIMULATION; BENCHMARKING; MODELING; PROTOTYPING; INSTRUMENTATION; TUNING
10	STRAINING SCIENCE CAPABILITIES	TECHNICAL ANALYSIS; COST-BENEFIT ANALYSIS; PROTOTYPING; REFERENCE CHECKING

Risk mitigation, prevention, Barry Boehm, USC

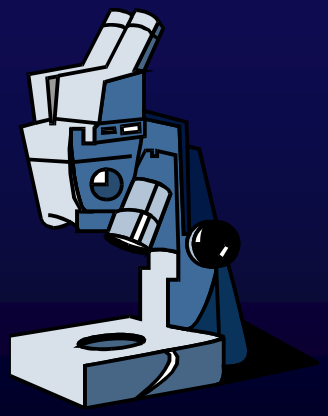
WIIFM - Stakeholders

● Managers ● Individual contributors ● Customers ● Owners



[JON1] Capers Jones, *Polishing the Software* Process, systems software, 1,000 FP, 125 KLOC, in C

Gaining support - 1



WHAT CAN YOU DO?

Donald E. Over

- Notorious micromanager.
- Wants to discuss requirements endlessly to ensure they are understood.
- Gets into design issues. Constantly asking how things are going to be done. Has some computer background.
- Wants daily reports on progress for a six month project.

Gaining support - 2

WHAT CAN YOU DO?

Ian DeLegate

- "What do you think?" is his favorite expression.
- He says that he gives people the freedom to succeed or fail on their own merits. He really means "read my mind".
- In the last meeting with he said, "I expect you and your staff to know what I mean, you're the experts in this field. That's why we came to you."

Gaining support - 3

WHAT CAN YOU DO?

Everett Bristle

- "How should I know." is a frequent statement (not a question).
- He really doesn't know.
- In fact, he couldn't know. He is breaking new ground with his request.
- He's defensive because he thinks he should know.
- In his division, he is the resident expert and is expected to (and does) have the answers.

Modeling "system" requirements

- Develop understanding
  - Simplify
  - Elaborate, refine
  - Eliminate ambiguity
  - Present, review
- Gain agreement with impacted parties
- Confirm with customer
- Joint development of requirements

Apply existing Engineering tools and techniques

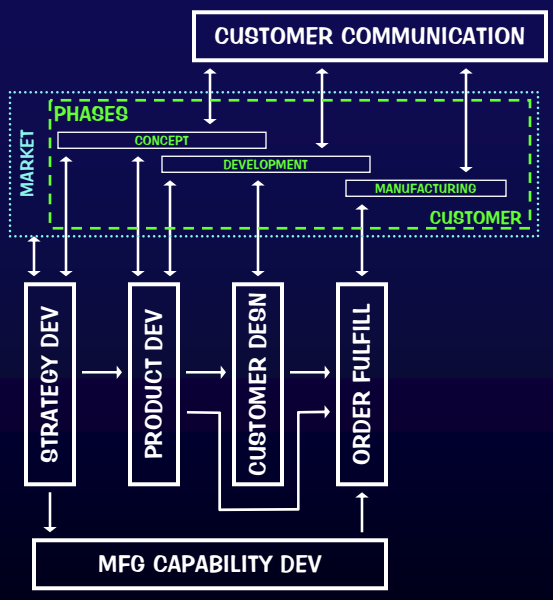
A picture is worth a thousand words
Fred R. Bernard, 1927

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

The benefits of graphical representation

The process map shows six processes for a $4 billion business. 'You know,' commented a TI executive, 'until we drew this picture ***we thought we were a lot more complicated than we really are***.'

Reengineering the Corporation, Hammer and Champy, p. 118



83

Techniques and Representations

Evolving systematically from (at least) 1979 (Demarco) to the unified modeling language (UML)

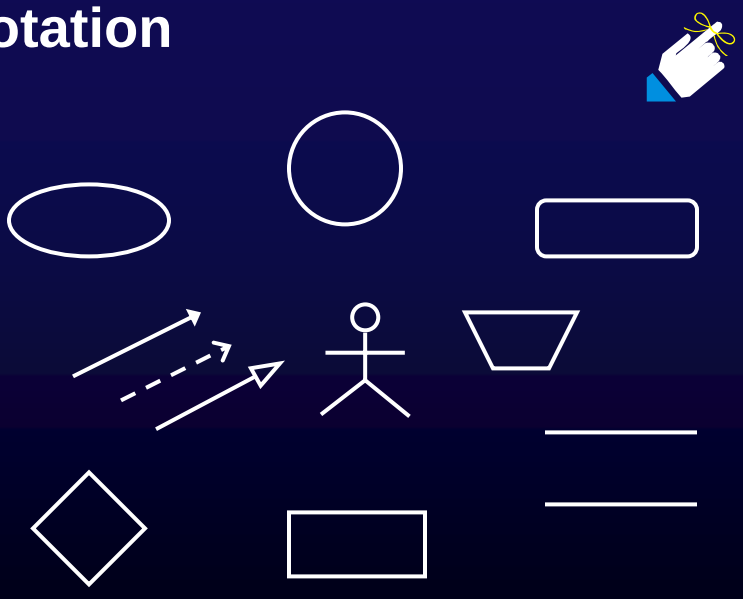
	FOCUS	METHOD
WHAT -	activities & deliverables real-world context	Data Flow Diagram (DFD) UML: Use Case Diagram
WHO -	organization & responsibility	Entity Relationship Diagram (ERD) UML: Collaboration Diagram
WHY -	cause & effect	State Transition Diagram (STD) UML: Statechart Diagram
HOW -	procedure & logic	Flow Chart (FC) UML: Activity Diagram, Sequence Diagram

84

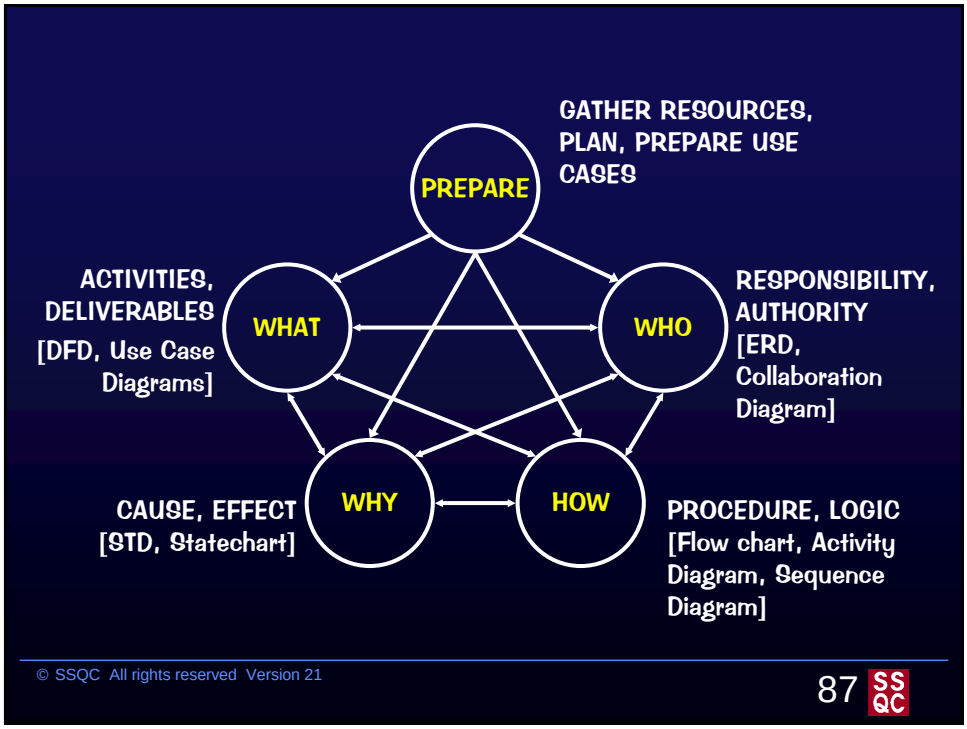
Tools

- General purpose presentation
  - Integrated (e.g., MS PowerPoint with live connectors)
- Diagramming
  - VISIO, allClear, Argo/UML, ...
- Process modeling and simulation
  - Process98, Optima, Argo/UML, etc.
- Flip chart, whiteboard, markers, ...

Notation



Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com



Prepare

(A) Identify representatives from affected groups

- Subject matter experts (SMEs)
- Knowledgeable - who do the work
  - From internal and external sources
- Interviewed or participate
- Buy-in, commitment, accuracy

Prepare (Cont.)

(B) Plan
- Estimate effort
- Schedule - committed resources
- Sponsor/customer and development management agreement

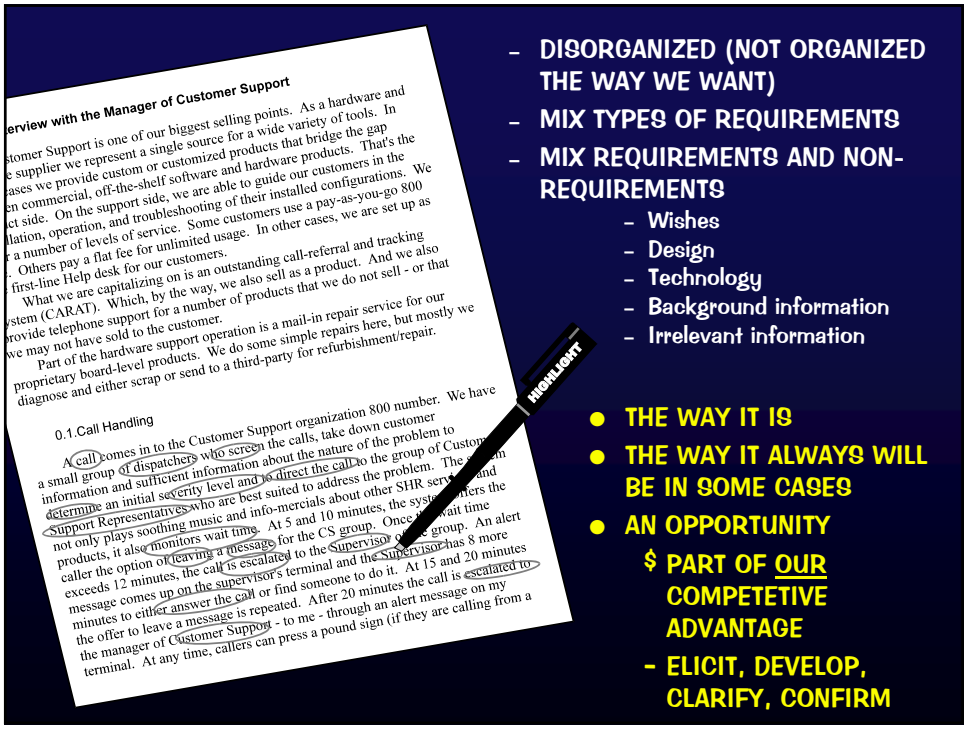
(C) Execute
- No observers
- Report progress - against a WBS
- Revise plan as required

Prepare (Cont.)

(D) Analyze or jointly develop customer input

ELEMENT	DESCRIPTION
WHY	Goal, purpose, impact, value
WHO	Participants, customers, suppliers, other systems
WHAT	Normal flow of work, outputs
WHAT ELSE	Exceptions, outputs and work flow
WHEN	Preceding events, triggers, inputs
WHERE	Operational context, environment, tools
HOW WELL	Performance criteria

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com



Engineering techniques

- Data flow diagram (DFD)
- Use case diagram
  - Include, extend
- Entity relationship diagram (ERD)
- Collaboration diagram
  - Parent, child
- STD
- Statechart diagram
- Flow chart
  - Deployment flow chart
- Sequence diagram
- Activity diagram
  - Fork, join

WHAT - Activities & Deliverables

- Customer- and supplier-centric view of process and activities
- Partition the process into activities
- Define relationships between activities and real world
- Identify
  - The services to be provided - outputs
  - The service required - inputs

WHAT - Activities & Deliverables (Cont.)

(A) Create a top-level DFD - single activity
(B) Decompose each activity in the DFD
(C) Repeat (B) as required

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

DFD Notation

EXTERNAL ENTITY - "SUPPLIER" OR "CUSTOMER"

FLOW OF WORK PRODUCTS AND INFORMATION

ACTIVITY, DOMAIN, CORE COMPETENCY

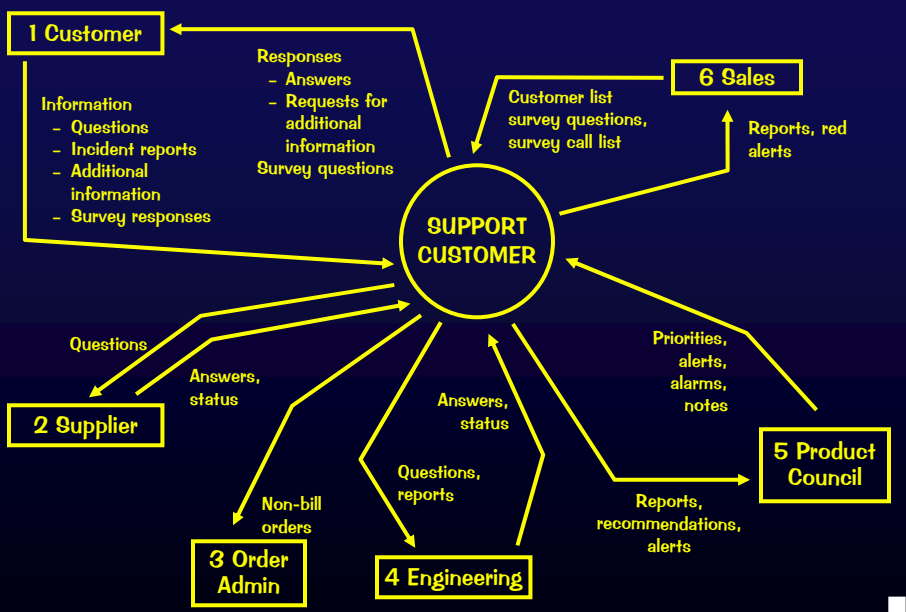
DATA STORE

WHAT - Activities & Deliverables (Cont.)

(A) create the top-level DFD
- Architecture
- The external
  - Deliverable(s)
  - Input(s)
  - Customers
  - Suppliers
- View process as a single activity - set the scope

Top-level DFD aliases
✔ Context diagram
• Level 0 DFD
DFD aliases
• Bubble chart
• Bubble diagram
• Process model
• Work flow diagram
• Function model
• Activity model

A Level 0 DFD - SHR Customer Support



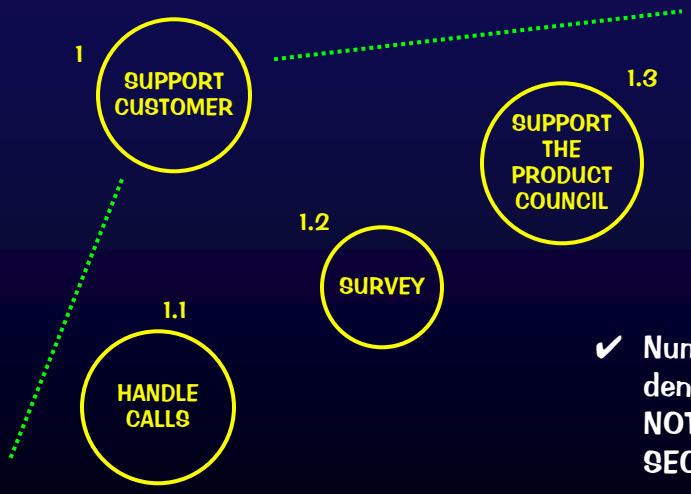
WHAT - Activities & Deliverables (Cont.)

(B) decompose each activity (bubble) in the DFD

(C) repeat (B) for each activity in the previous level DFD. Continue until sequential steps are reached.

The DFD is not a flow chart.

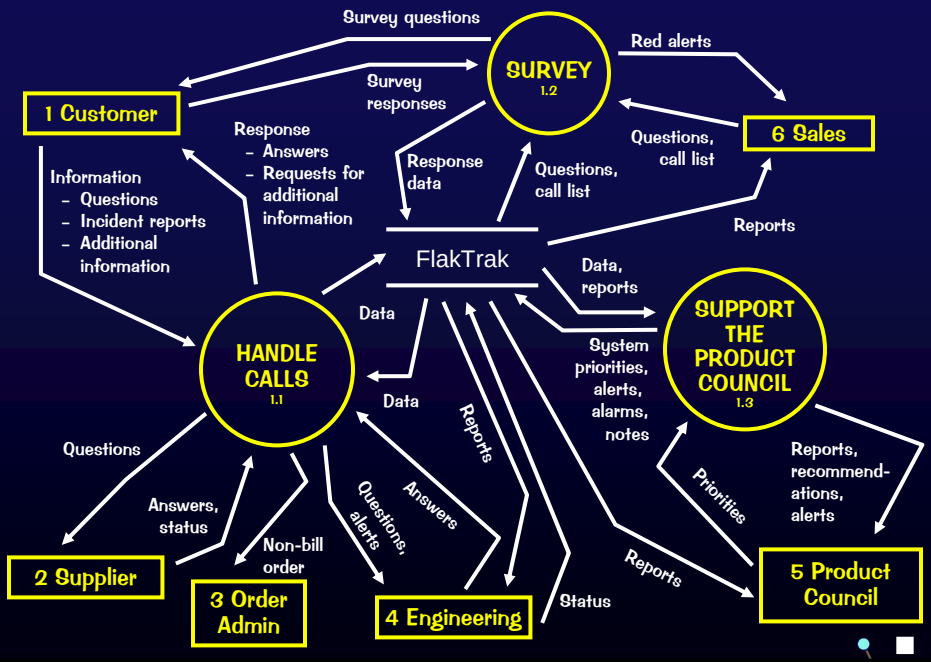
Creating a Level 1 DFD for SHR Customer Support



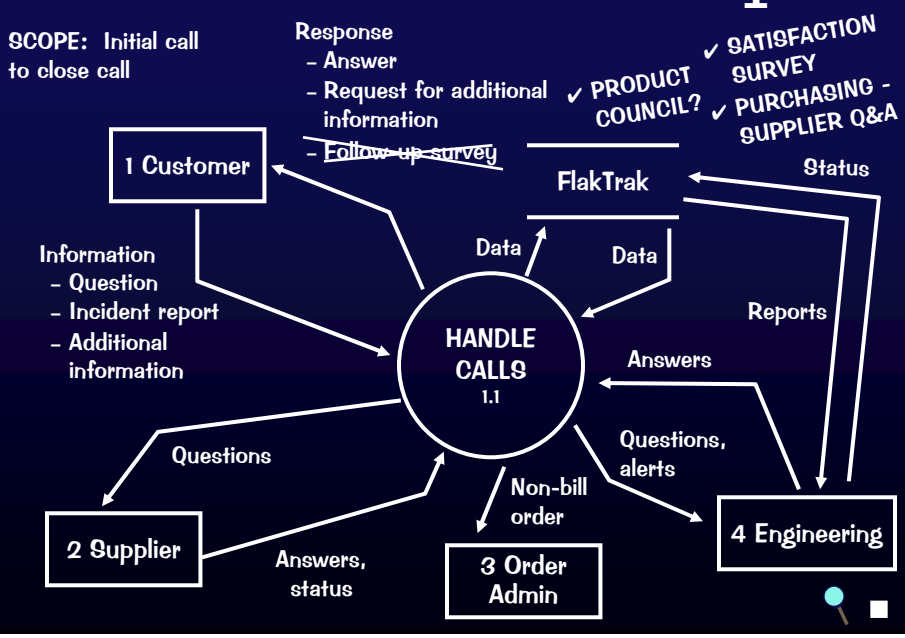
✔ Numbers denote level - NOT SEQUENCE

99

A Level 1 DFD - SHR Customer Support



What Actually Happened - Level 0 DFD 1



WHAT - Activities & Deliverables (Cont.)

RULE 1
Preserve inputs

RULE 2
Preserve outputs

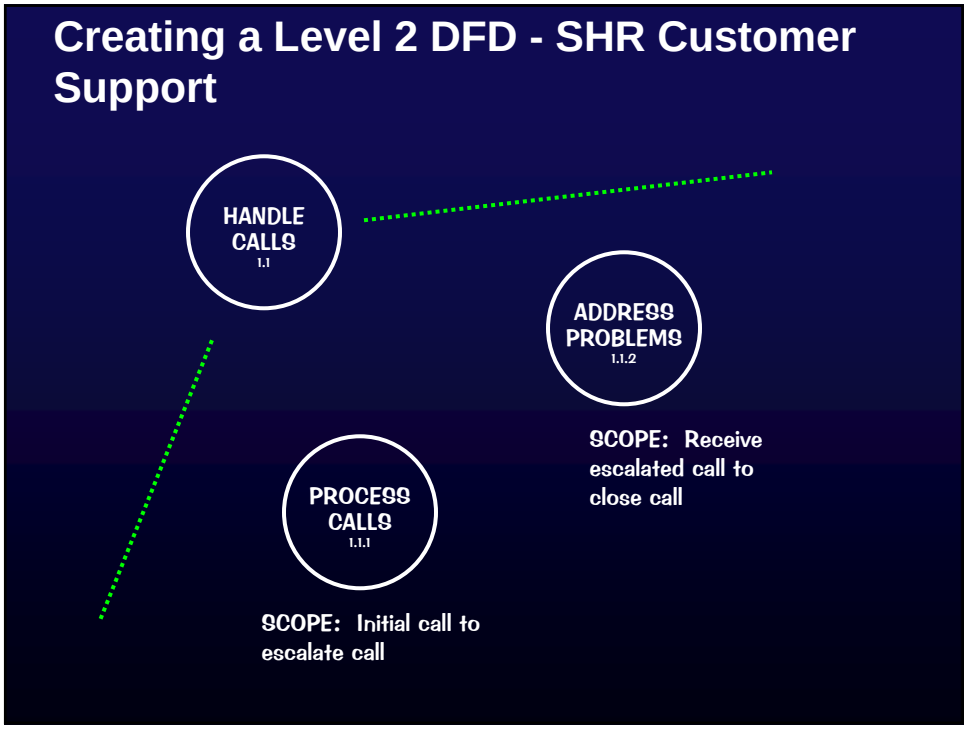
RULE 3
Continue until sequential steps are reached. The DFD is not a flow chart.

RULE 4
Derive a maximum of 9 bubbles from each higher level bubble

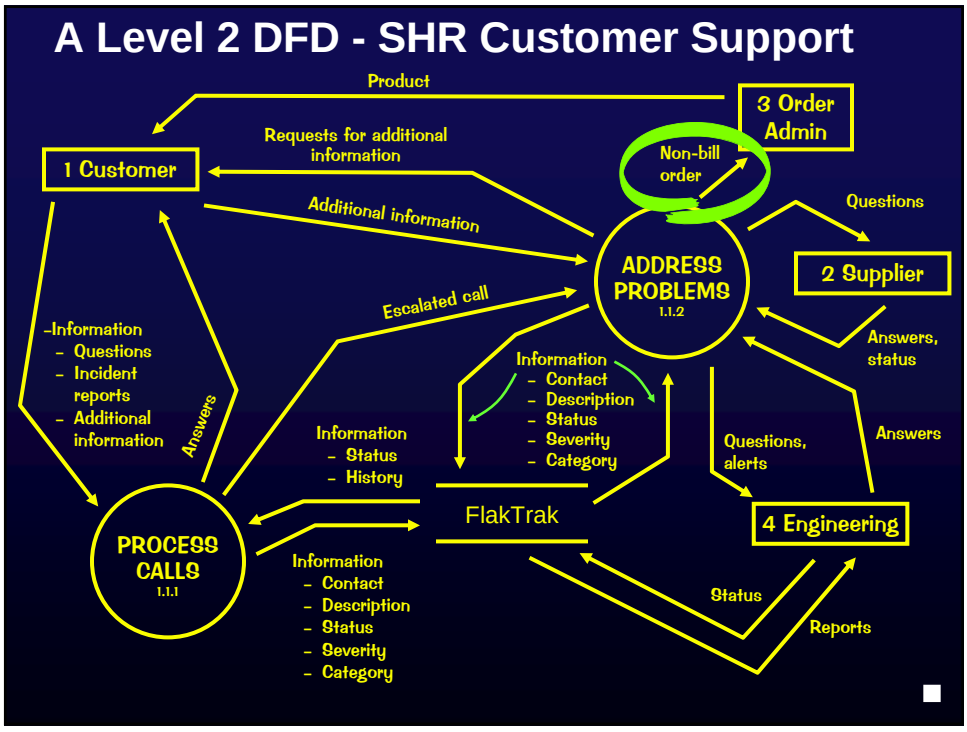
COROLLARIES
- ✔ You will have to back track
- ✔ Use subteams based on the scope of the activity

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Creating a Level 2 DFD - SHR Customer Support



A Level 2 DFD - SHR Customer Support



A DFD Captures

- Activities
- Customers
- Suppliers
- The flow of deliverables between activities - inputs and outputs

A DFD does NOT capture

- Time
- Responsibility
- Logic

Health Monitoring System (HMS)

The vital signs of high-risk patients at Pinafore Regional Hospital (PRH) are closely monitored. Pulse, temperature, and blood pressure sensors on the patient are read by HMS every 30 seconds. HMS records actual readings in the patient's record and compares them to ranges specified for the particular patient by the assigned doctor.

If a value falls outside the range, HMS records the exception in the patient's record and sends an alert to the terminal at the appropriate nursing station so an appropriate response can be initiated.

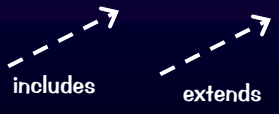
Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

The UML: Use Case Diagrams

Notation

USE CASE - A COHERENT UNIT OF FUNCTIONALITY PROVIDED BY A SYSTEM INVOLVING ONE OR MORE ACTORS

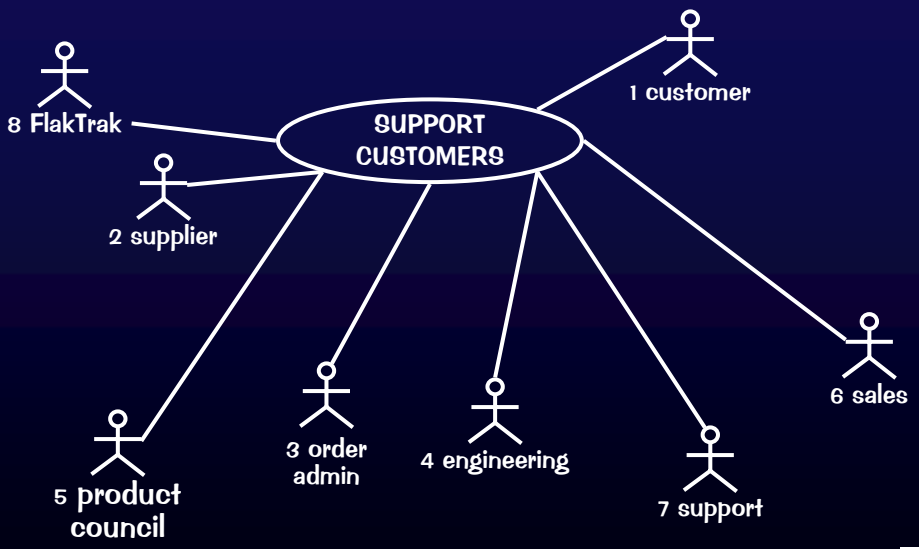
ACTOR - A ROLE PERFORMED BY AN OBJECT IN INTERACTING WITH THE SYSTEM - CUSTOMER [RECEIVES VALUE], SUPPLIER [PROVIDES INPUT], PARTICIPANT, DATA STORE



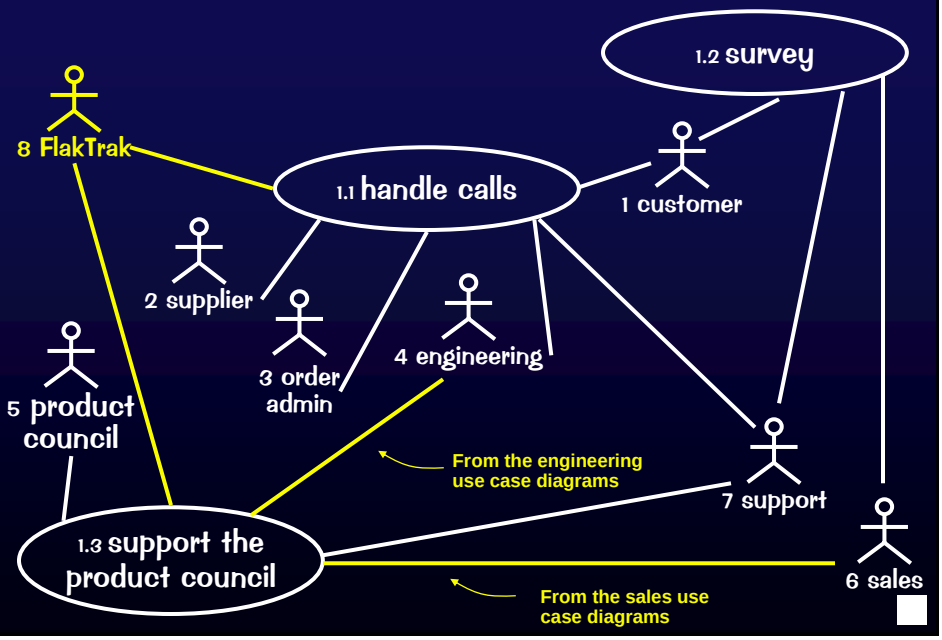
JOIN USE CASES

JOINS AN ACTOR AND A USE CASE

"Level 0" UCD - SHR Customer Support



"Level 1" UCD - SHR Customer Support



Optimizing the Set of Activities

- Identify relationships among activities
  - Maximize standardization
    - Aggregate related activities
    - «extends»
  - Minimize duplication
    - Isolate common activities
    - «includes»

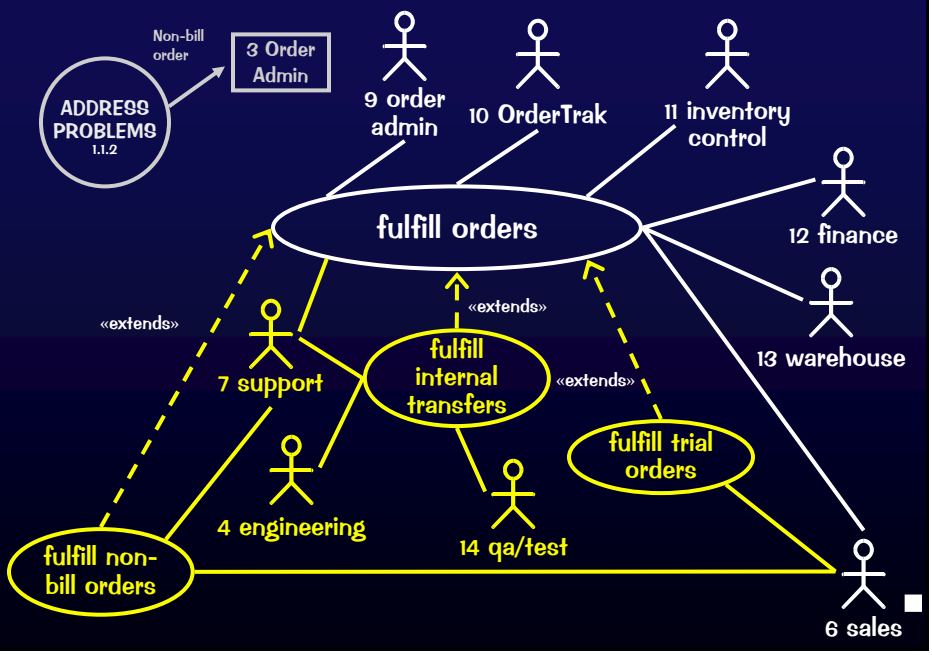
Expediting a rush order and rejecting an order are variations on processing a standard order.

Processing an inventory transfer and a requisition both require that the approver verify the budget balance.

Relationships Between Activities: «extends»

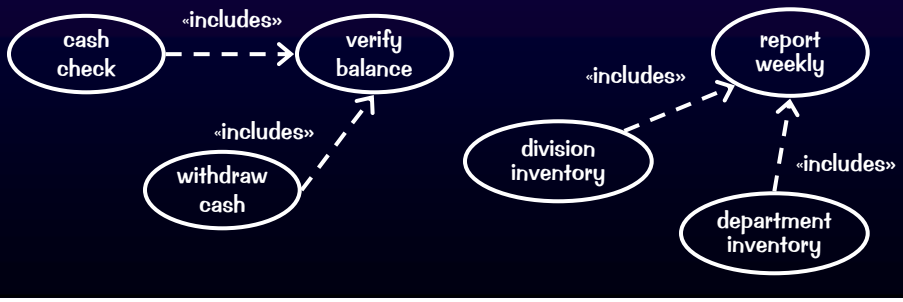
- Under certain conditions, Activity B includes the behavior in Activity A (e.g., exception handling)
- Activity A is an optional part of Activity B
- Activity A is a variant of Activity B

Relationships Between Activities: «extends»



Relationships Between Activities: «includes»

Activity C includes Activity D.



WHAT - Activities & Deliverables - Summary

- Identify and optimize

DFD
- Activities
- External customers
- External suppliers
- Data store
- The flow of deliverables between activities

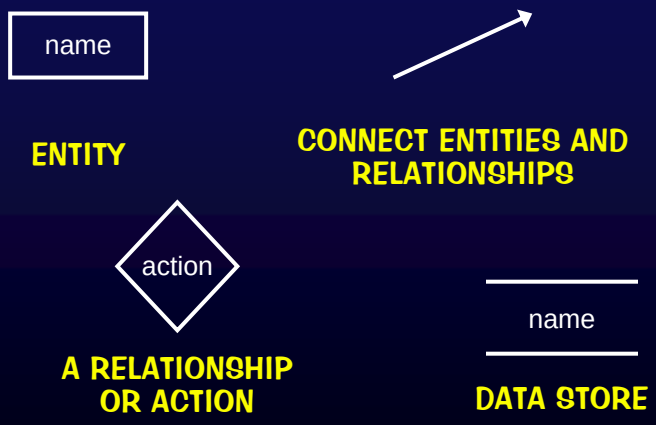
The UML Use Case
- Participants
- Family relationships among activities

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

WHO - Responsibility & Authority

- Entity = organizations, people, deliverables, etc.
- Identify and define the relationships and interactions among the entities associated with the process
- Entities: from the nouns in the narrative
- Relationships: from the verbs

Entity Relationship Diagram (ERD) Notation



For Example

Relationships may be:

- One to one
- Many to one
- One to many
- Many to many
- None to one

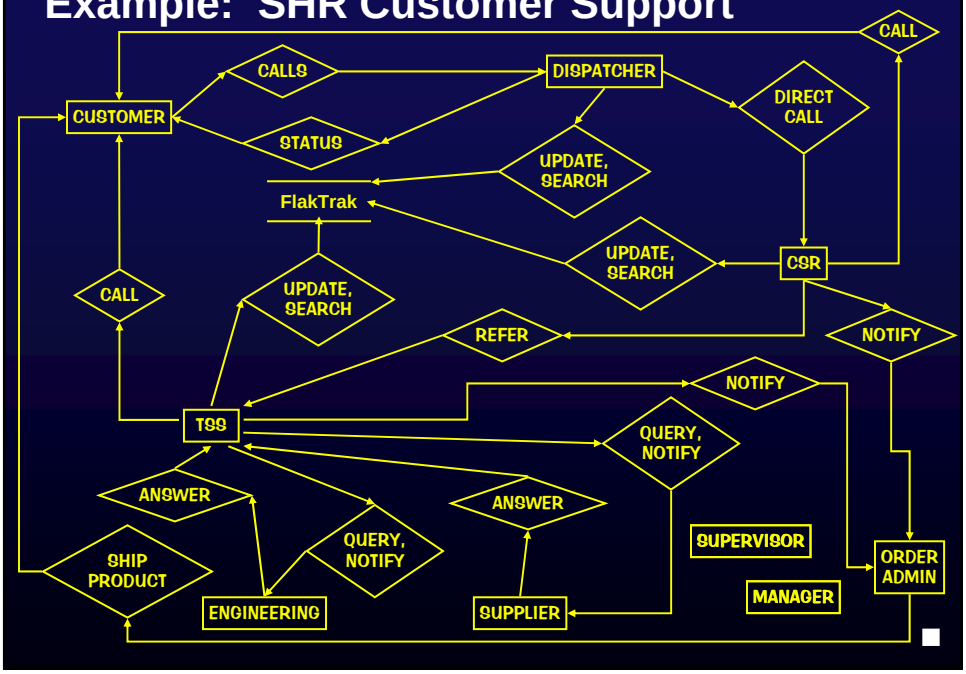
Rules for Appearing in an ERD

1 Represent “things” in the real world
- People, products, reports, strategies, standards, ...

2 When the entity is a set of objects, there is some way to distinguish individual objects
- For example: customers, items

3 Each entity is necessary to the process
- For example: supervisor monitoring
- Consistent with DFD

Example: SHR Customer Support



An ERD captures

- Actions, interactions
- Responsibility, relationship

An ERD does NOT capture

- Time
- Logic

Health Monitoring System (HMS)

The vital signs of high-risk patients at Pinafore Regional Hospital (PRH) are closely monitored. Pulse, temperature, and blood pressure sensors on the patient are read by HMS every 30 seconds. HMS records actual readings in the patient's record and compares them to ranges specified for the particular patient by the assigned doctor.

If a value falls outside the range, HMS records the exception in the patient's record and sends an alert to the terminal at the appropriate nursing station so an appropriate response can be initiated.

In some cases, the doctors specify the ranges in written orders which are entered into the system by the nurses.

The nurses also check patients periodically - both in person and by inspecting the values recorded in HMS.

If a sensor reading goes to zero, the nurses' station is notified.

The UML: collaboration diagrams

Notation - fewer lines, structured notation, sequence

AN OBJECT THAT INTERACTS WITH THE SYSTEM - CUSTOMER, SUPPLIER, PARTICIPANT, DATA STORE

PATH BETWEEN OBJECTS

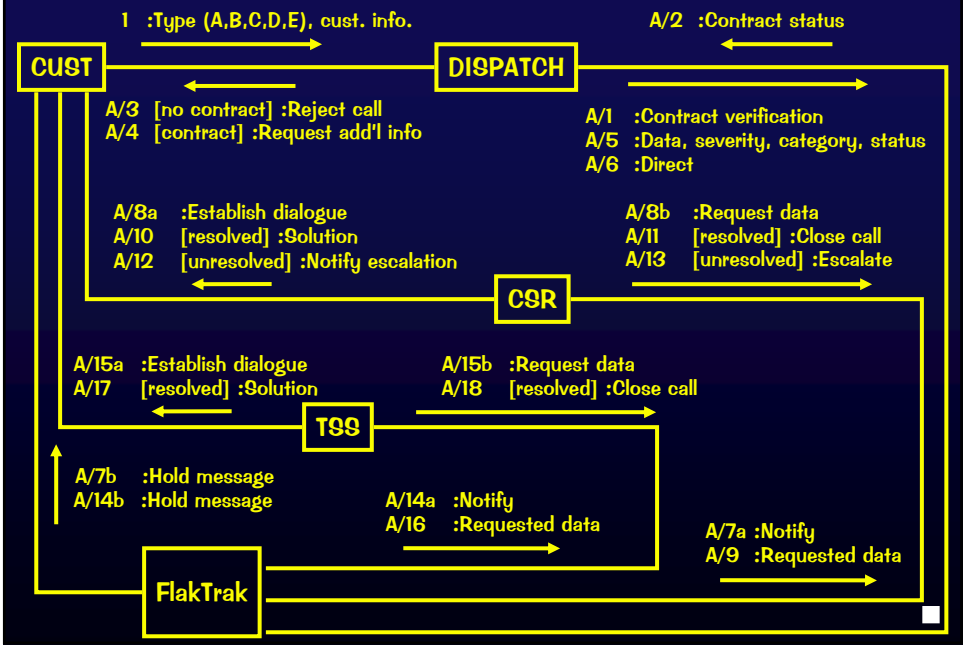
FLOW

label message

A DESCRIPTION OF AN INTERACTION BETWEEN OBJECTS ALONG A PATH

JOIN OBJECTS AS PARENT AND CHILD

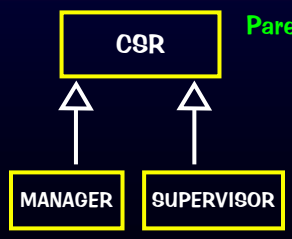
Example: SHR Customer Support



Optimizing interactions

- Identify relationships among objects
  - Maximize standardization
  - Minimize duplication
  - Generalization - *parent* and *child*

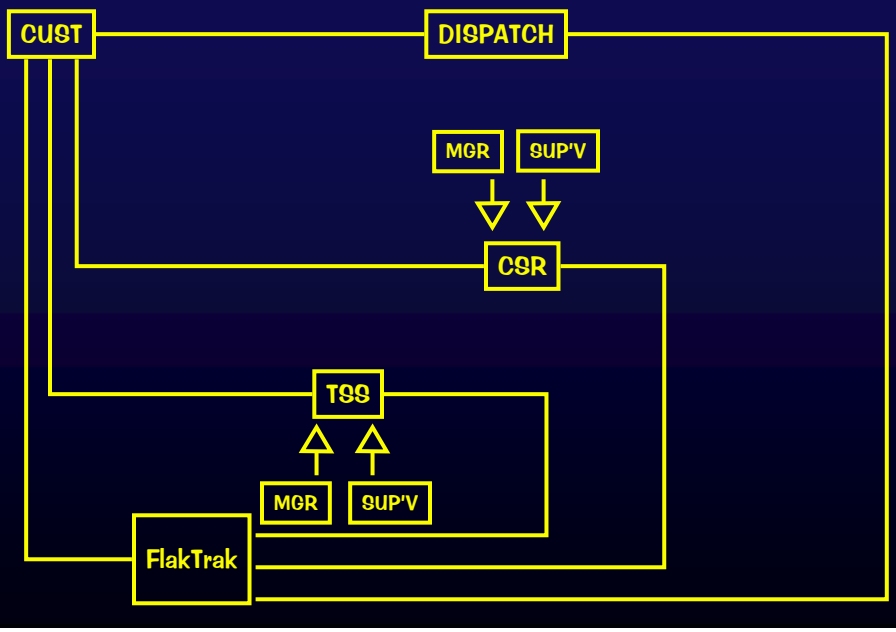
- The child inherits all of the responsibilities of the parent.
- The child may be substituted for the parent.
- The child may add to or modify responsibilities inherited from the parent.



A Supervisor or Manager may be called on to act as a CSR.

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Example: SHR Customer Support



WHY - cause & effect

- Identify the different states the PROCESS may be in
  - Event-driven activities
  - Not work product states
  - *ON HOLD* versus *INVESTIGATING*
- List the triggering event and the resulting action as annotation

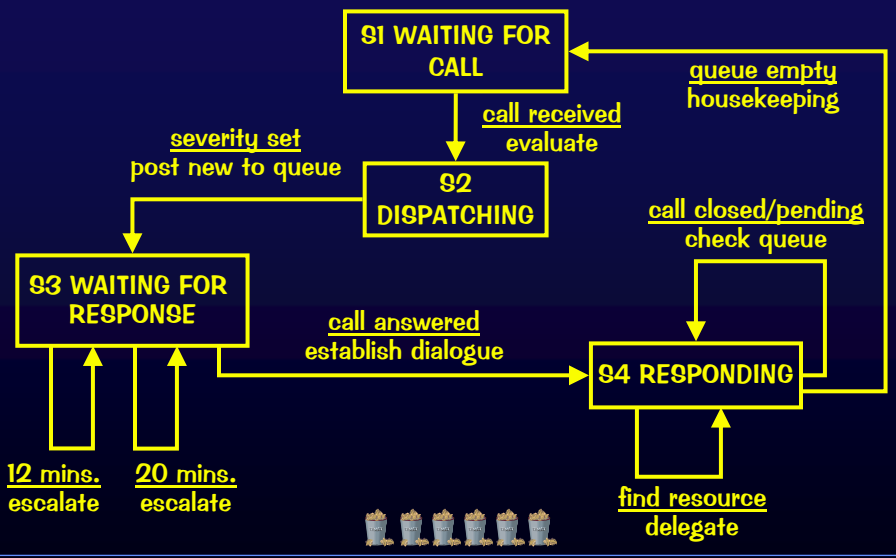
State Transition Diagram (STD) notation

triggering event

resulting action

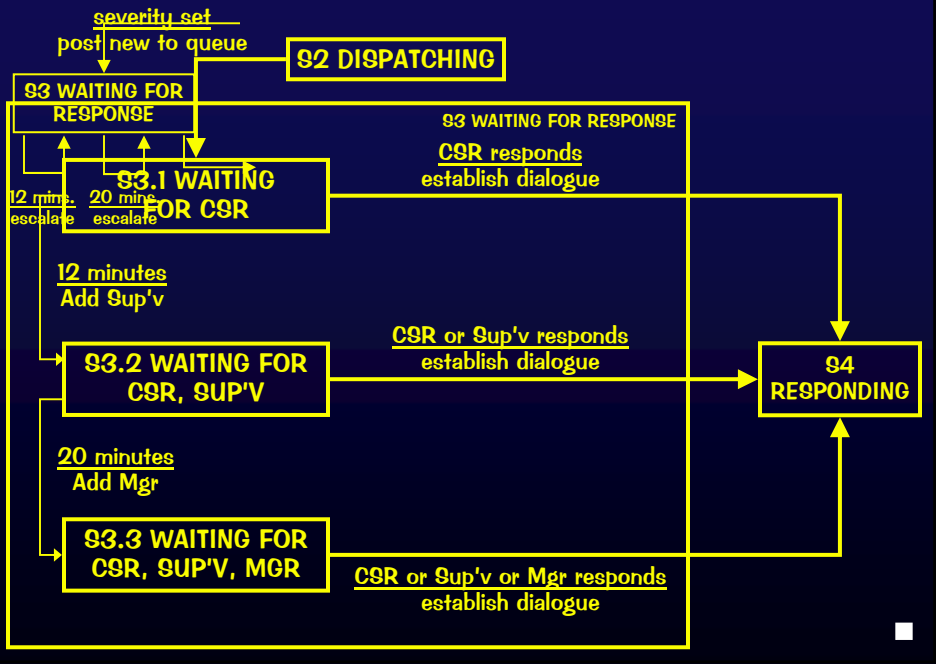
Annotation required on each arrow

Example: SHR Customer Support



Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Example: SHR Customer Support (cont.)



Rules for an STD

RULE 1

Every state except the initial state must have a predecessor state

RULE 2

Every state except the final must have a successor state

RULE 3

Every state change must have a triggering event and resulting action

Health Monitoring System (HMS)

The vital signs of high-risk patients at Pinafore Regional Hospital (PRH) are closely monitored. Pulse, temperature, and blood pressure sensors on the patient are read by HMS every 30 seconds. HMS records actual readings in the patient's record and compares them to ranges specified for the particular patient by the assigned doctor.

If a value falls outside the range, HMS records the exception in the patient's record and sends an alert to the terminal at the appropriate nursing station so an appropriate response can be initiated.

In some cases, the doctors specify the ranges in written orders which are entered into the system by the nurses.

The nurses also check patients periodically - both in person and by inspecting the values recorded in HMS.

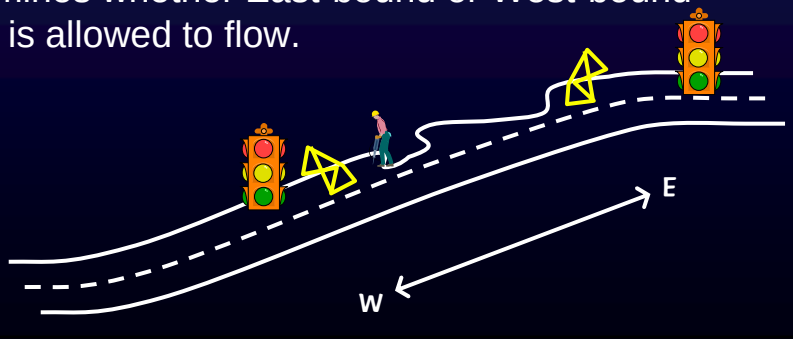
If a sensor reading goes to zero, the nursing station is notified.

If an alert is not acknowledged from the nursing station within 30 seconds, the nursing supervisor is paged with an appropriate action code, which indicates the source of the alert.

131 SSQC

Under repair

A section of a two-lane, two-way highway is under repair. As a result only one lane is open. Traffic lights and sensors are set up at either end of the one-lane section. The sensors and lights are connected to a controller, which determines whether East-bound or West-bound traffic is allowed to flow.



An STD captures

- Relative time, sequence
- Triggering events and initial actions
- Control, alternatives

An STD does NOT capture

- Logic
- Flow of deliverables
- Responsibility, relationship

The UML Statechart notation

A state

A permissible state change

'[' triggering event ']' '/' action-expression

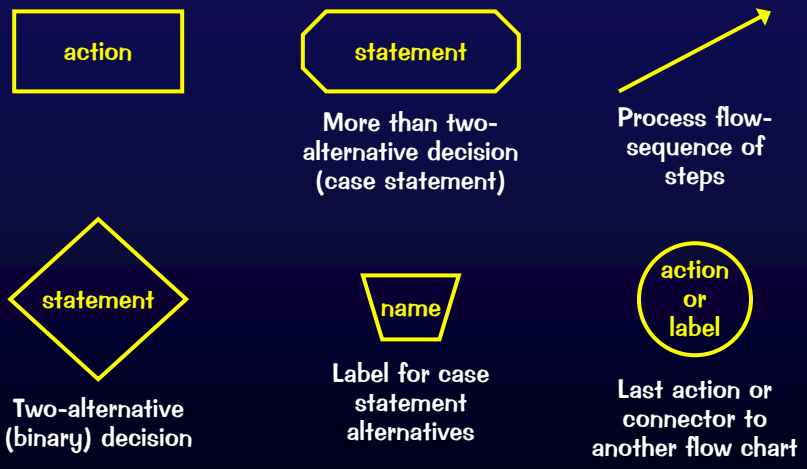
Annotation on each arrow

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

HOW - procedure & logic

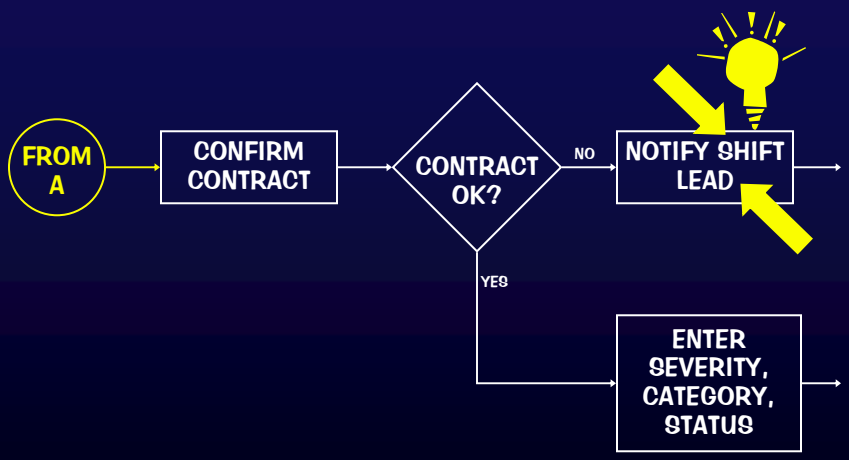
- Model processes in which transitions between steps are triggered by the completion of other activities
- Start with a single box
- Define first and last steps

Flow chart notation



Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

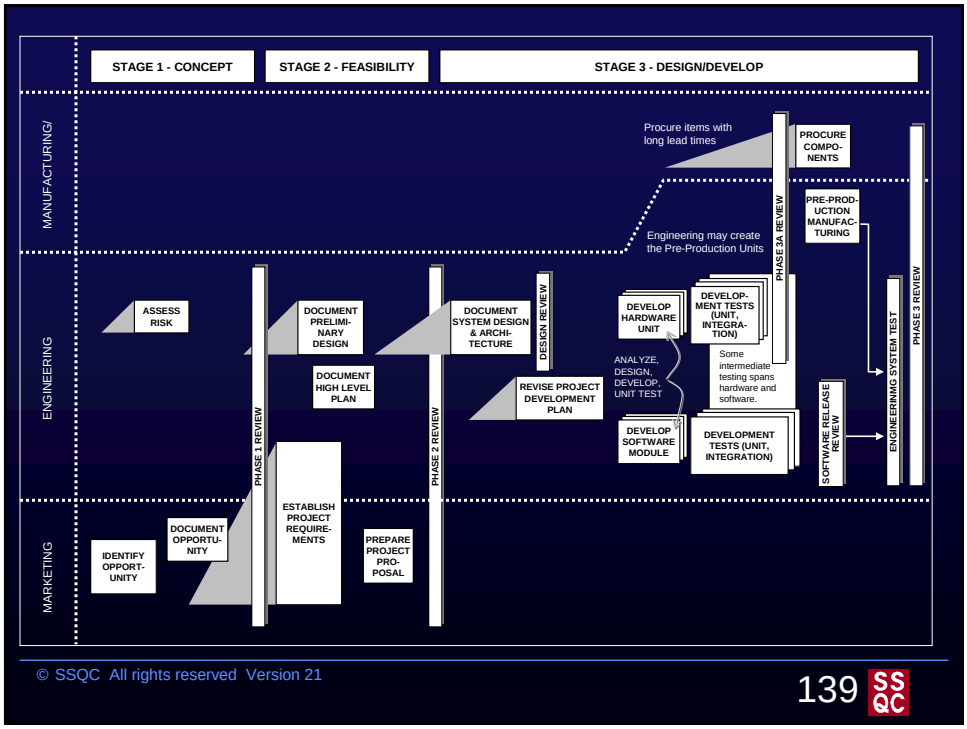
Surprise!

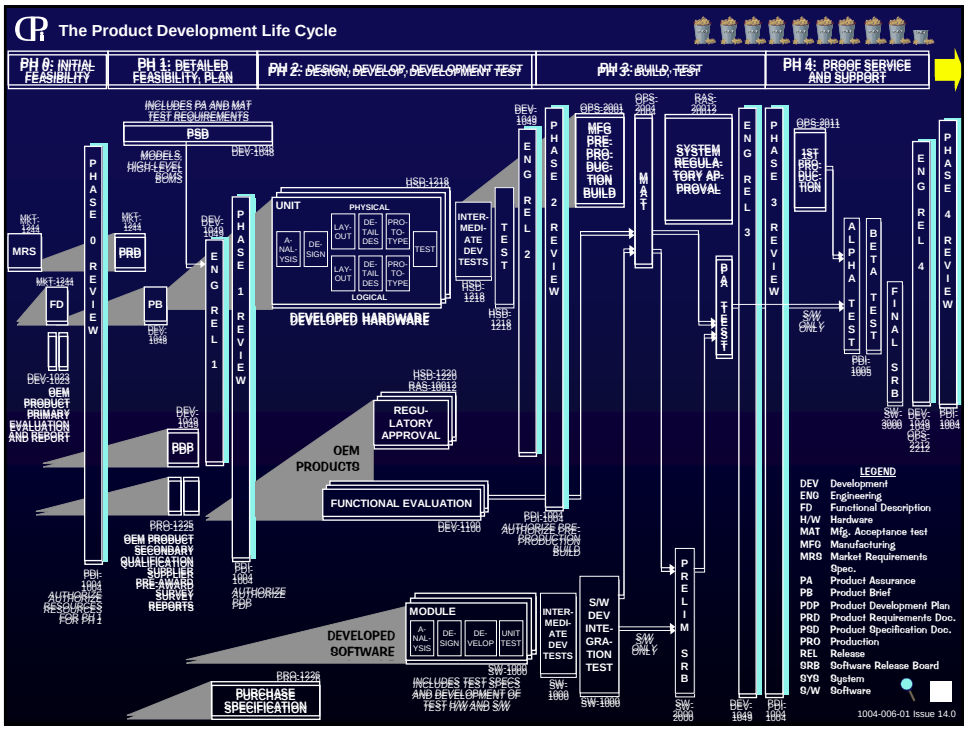


How detailed? Depends on ...

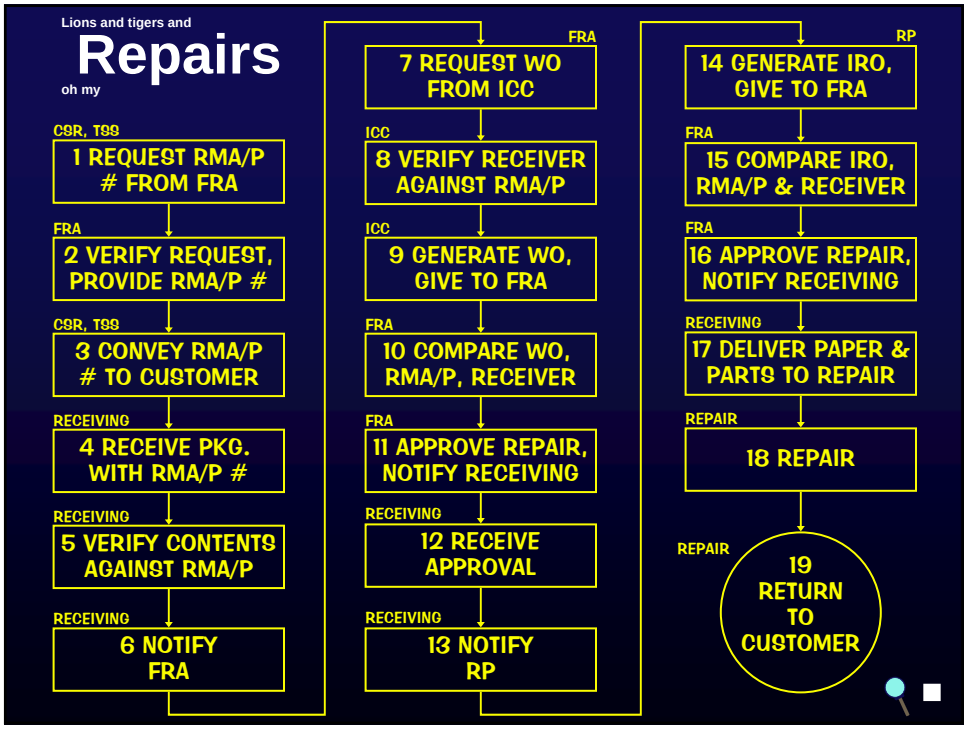
- Where you start
  - Stability
  - Complexity
  - Knowledge, experience, education
- How they will be used
  - Throw away
  - Part of the contract
  - Requirements documentation

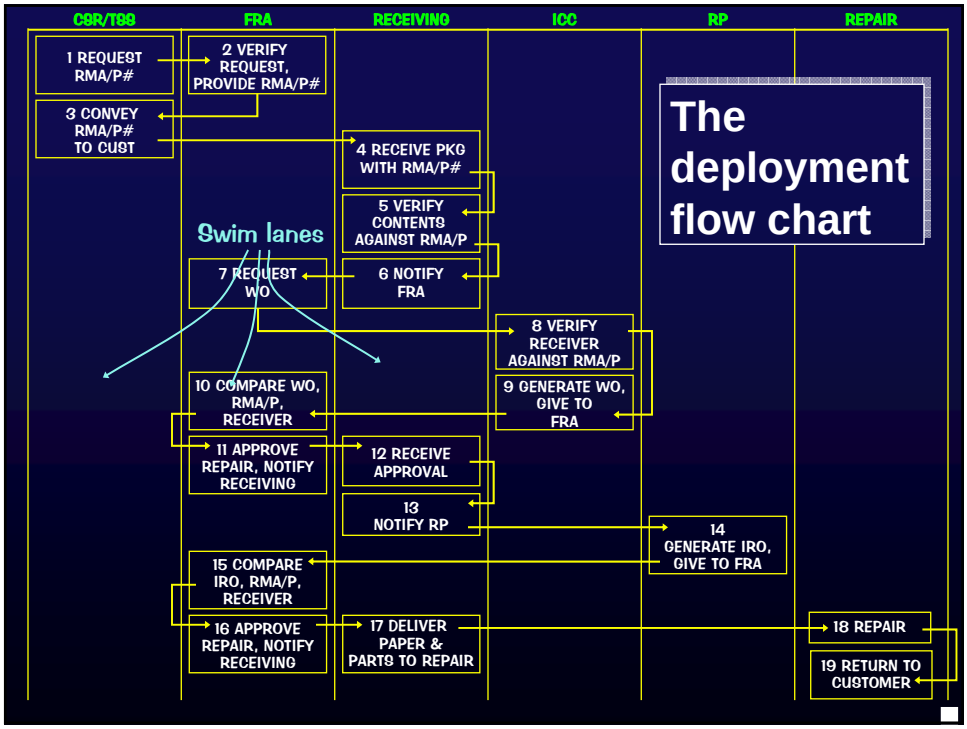
Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com





"more complicated than we really are"





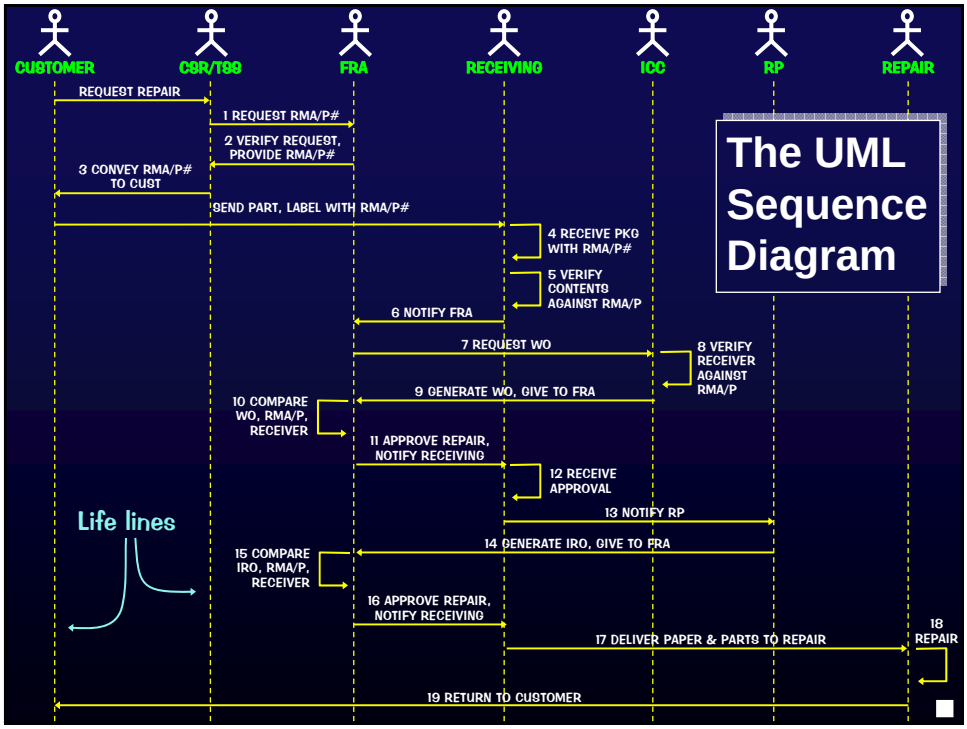
A flow chart captures

- Actions
- Responsibility
- Logic and rules/control
- Time - cause and effect

A flow chart does not easily capture

- Parallel flows
- Complex flows, numerous alternative routes and decision criteria, numerous entry points

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com



The UML Activity Diagram

Notation

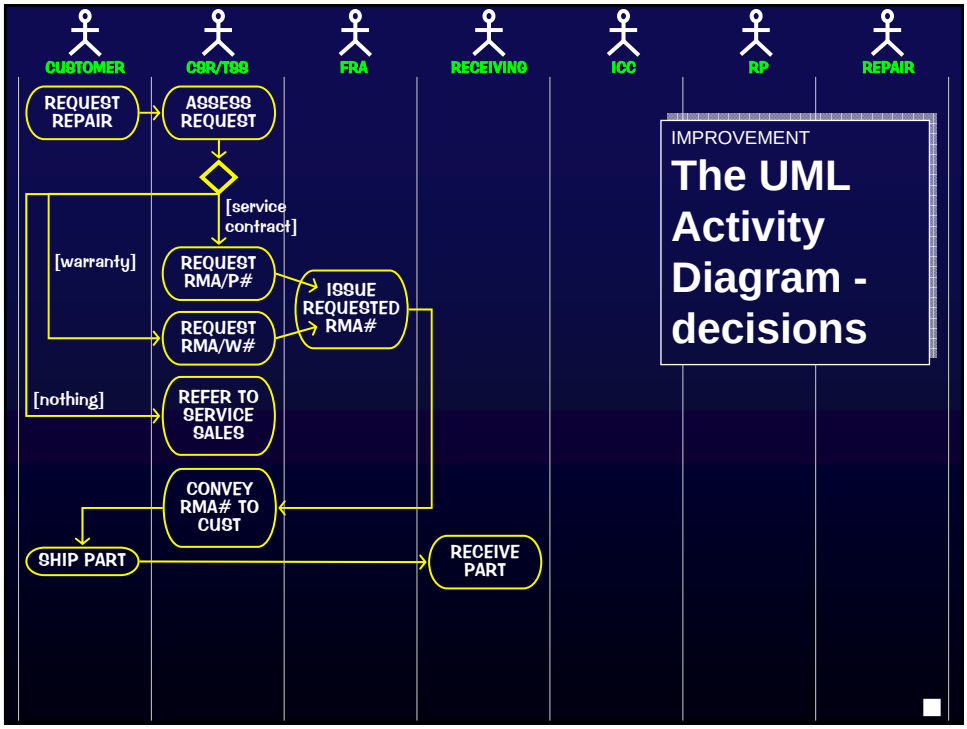
activity

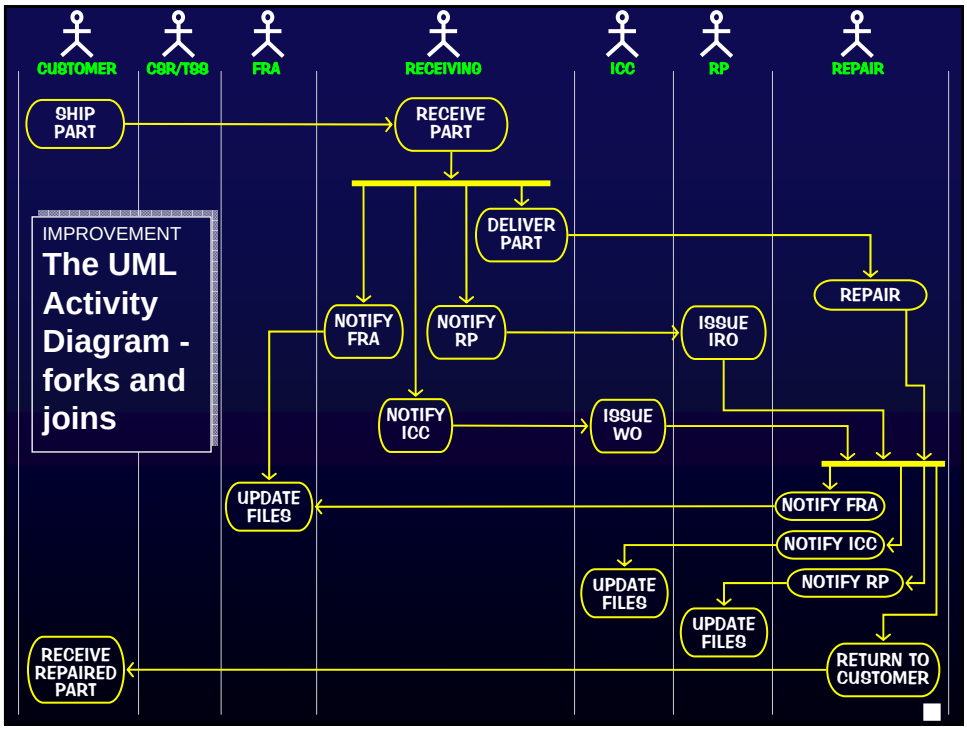
Fork - beginning of parallel activities

Process flow - sequence of activities

Decision point (any number of alternatives)

Join - completion of parallel activities





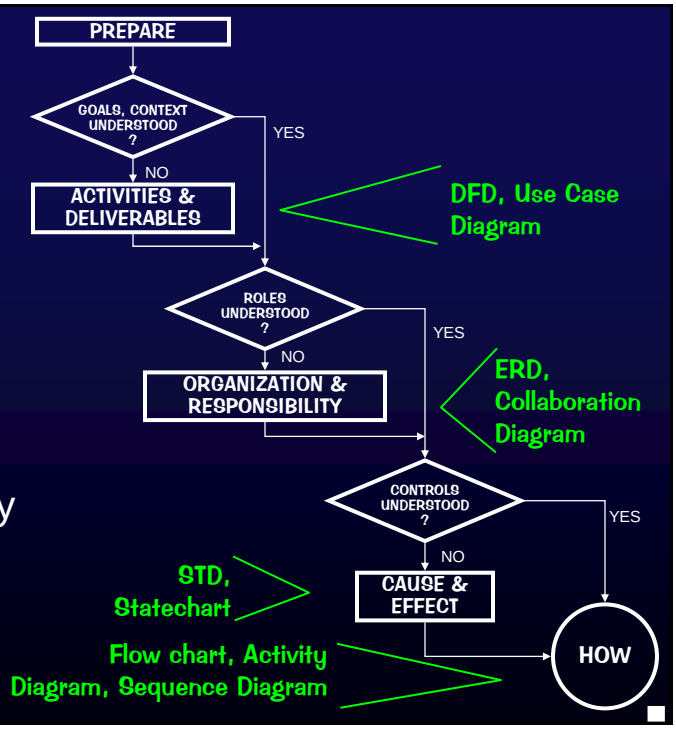
Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Confirm

For flow charts, Activity Diagrams, Sequence Diagrams

- Observe (show me vs. tell me)
- Walk through
- Review - present to others
- Ensure process capability is consistent with customer requirements

A suggested hierarchy



- Iterative - reenter at any point
- Optional steps

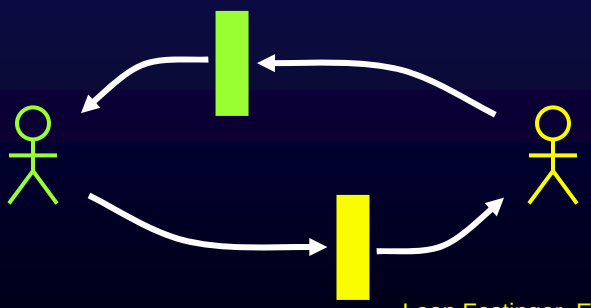
Juran's six sources of error

- Misinterpretation
- Inadvertent errors
- Lack of technique
- Conscious errors
- Bias
- Futility

151

Selective listening, blind spots, and other self-inflicted limits

- Cognitive dissonance
- Rational versus rationalizing



Leon Festinger, Elliot Aronson

152

FINAL TEAM EXERCISE

Assignment

Prepare and deliver a 10 minute presentation to H. Strasser and B. Arnold. Your goal is to **KICK OFF** the requirements-gathering phase of a possible project so focus on requirements and questions from the memorandum.

Suggested Topics

- Introductions
- Summarize input (confirm understanding)
- Identify basic requirements
- List issues and questions for follow-up

A Test Drive

SHR INCORPORATED

Formal Input Tracking System **FITS**

		FITS Identifier **FID**	1204-006-030422.183-0454Z
	UNIVERSAL PERSONNEL CODE UPC	**NAME**	**LOCATION**
SHR CONTACT	199046134	Delane, E.	CODE xx DEPT yy

DISPOSITION	**REQUIREMENT STATEMENT**
X Defect Change Close-SOAD Investigate	When an inter-depot transfer is approved, it is charged against the depot budget and is correctly reported in the Depot Working Budget Report (DWBR-0021). When a partial shipment is received for an approved inter-depot transfer, the line item disappears from the Depot Working Budget Report and is not reflected in the totals. When the transfer is complete, the line item reappears and the totals are correct. (SEE ATTACHED SAMPLE REPORTS, DB DUMP) DWBR-0021 shall correctly reflect approved inter-depot transfers for which partial shipments are received.

IMPACT	This problem can affect month-end reconciliation since 1) an inter-depot transfer for which partials have been received may never be completed or it may extend over a month-end balance 2)approved inter-depot transfers for which partial shipments have been received are in the Regional Monthly Roll-up Report (REG-0033)

	CRITICAL	**IMPORTANT**	**ROUTINE**
WORKING PRIORITY	X	X	

Tel +1-408-985-4476
FAX +1-408-248-7772
info@ssqc.com www.ssqc.com

Contact Information

Bill Deibler, Bob Bamford
Software Systems Quality Consulting
2269 Sunny Vista Drive
San Jose, CA 95128
Phone 408-985-4476 Fax 408-248-7772
deibs@ssqc.com, bamf@ssqc.com

info@ssqc.com
www.ssqc.com

What Can Engineers and Project Managers Do about Requirements?

Gaining Support

Customer	Strategy
(1) Donald E. Over	
(2) Ian DeLegate	
(3) Everett Bristle	

NOTES

Test Drive

From the Formal Input Tracking System (FITS), you have the following information.

		FITS Identifier **FID**	1204-006-030422.183-0454Z
	UNIVERSAL PERSONNEL CODE (UPC)	**NAME**	**LOCATION**
ORIGINATOR	015477408	Skilling, P. W.	SM2-06 DEPT cc
	PRODUCT IDENTIFIER PID		**VERSION**
SYSTEM INFORMATION	Operating Budget Target Tracking Application - OBTTA		5.01.0004
PROBLEM OR REQUEST X Problem Request	The Extended Money Values for approved Inter-Depot Transfer Transactions are not permanently reflected in the on the Depot Working Budget Report. I reported this problem t/r #552950 which was cancelled w/o explanation. This is critical since depot managers use the budget report when making approval decisions.		
IMPACT	Depot managers may approve requests when budget not there.		
	CRITICAL	**IMPORTANT**	**ROUTINE**
ORIGINATOR PRIORITY		X	

Background

You download his data and confirm that Skilling has the correct versions of OBTTA and all the associated hardware and software..

You check FITS and determine that:

- TR 552950 was cancelled; the reason was "system operating as designed". The person who closed it is not available.
- You can't find another problem report related to the problem described.

You walk down the passageway to the Standard Systems Simulation Lab (S3L), load version 5.01.004 of OBBTA and Skilling's data, and determine that:

- A few sample inter-depot transfers you enter show up correctly in the Depot Working Budget Report (DWBR-0021).

Finally, because you are thorough, you accelerate time by a factor of 1,000 to simulate running the system for a month and determine that:

- The sample inter-depot transfers still show up correctly in DWBR-0021.
- The sample inter-depot transfers show up correctly in the Regional Monthly Roll-up Report (REG-0033)

Attributes	Types	(C1) Notes/Issues
❑ Clear ❑ Correct ❑ Consistent ❑ Singular ❑ Testable, verifiable ❑ Feasible ❑ Design independent ❑ Traceable ❑ Clear	❑ Functional (user, mode) ❑ Exception handling (user, mode) ❑ Performance (quantitative) ❑ Design constraint (environment) ❑ Interface (interact with user, external components and other internal components) ❑ Communications (connectivity, access) ❑ Computer security and access ❑ Backup and recovery ❑ Implementation (conversion, installation, hardware acquisition) ❑ Information (examples, other)	

(C2) Question/Topic		(C3) Notes/Answer
		
	STOP here until directed to continue to the next part.	

Test Drive - Capture the Requirements

(C4) Restated problem(s) or request(s)	(C5)
Restated impact(s)	

NOTES

Test Drive 1

From a request submitted for Flak-Trak, an automated hot-line system. Flak-Trak is installed in 7x24 help-desk centers that handle customer telephone calls.

> **After 12 minutes we need FlakTrak to automatically escalate any incoming call that has not been answered to the hot-shift line lead.**

Background

FlakTrak incorporates a standard HoldAll Call Director, which automatically places incoming calls on hold and plays soothing music until they are answered by a support person. FlakTrak includes software that polls the HoldAll twice every second and which displays a constant on-screen message at all active support work stations regarding how many calls are waiting in the queue. For each call, FlakTrak also records the date, time, sequential identification number, and the amount of time between first ring and when the call is answered. The information is recorded in a file that can be read in Microsoft Excel.

Attributes	Types	(C1) Notes/Issues
❑ Clear ❑ Correct ❑ Consistent ❑ Singular ❑ Testable, verifiable ❑ Feasible ❑ Design independent ❑ Traceable ❑ Clear	❑ Functional (user, mode) ❑ Exception handling (user, mode) ❑ Performance (quantitative) ❑ Design constraint (environment) ❑ Interface (interact with user, external components and other internal components) ❑ Communications (connectivity, access) ❑ Computer security and access ❑ Backup and recovery ❑ Implementation (conversion, installation, hardware acquisition) ❑ Information (examples, other)	

(C2) Question/Topic		(C3) Notes/Answer
		
	STOP here until directed to continue to the next part.	

Test Drive - Capture the Requirements

(C4) Restated problem(s) or request(s)	(C5)
Restated impact(s)	

NOTES

Test Drive 2

From a bug report for an inventory management system your group maintains. The report is listed as originated by M. Wagner at the Denver Depot.

> **When the Flash utility is used to add and item to the Depot Parts List (DPL), the item doesn't show up in a DPL Query. But when DPL Add is run, it says the item is already there. If an item is added with DPL Add everything works fine. Except Flash overwrites an item with the same part number.**

Background

DPL is an application that incorporates a commercial data base. DPL Add and DPL Query are tools provided in the DPL application suite (along with DPL Delete, DPL Edit, DPL Report, and DPL Analyze). Flash is a $29.00 commercial utility program that provides direct access to the database for users to delete, edit, and add records. Flash was distributed two years ago when the DPL application suite was discovered to be incompatible with the latest version of Windows, a problem that has since been corrected. You can duplicate the facts in the Problem Report.

Attributes	Types	(C1) Notes/Issues
❑ Clear ❑ Correct ❑ Consistent ❑ Singular ❑ Testable, verifiable ❑ Feasible ❑ Design independent ❑ Traceable ❑ Clear	❑ Functional (user, mode) ❑ Exception handling (user, mode) ❑ Performance (quantitative) ❑ Design constraint (environment) ❑ Interface (interact with user, external components and other internal components) ❑ Communications (connectivity, access) ❑ Computer security and access ❑ Backup and recovery ❑ Implementation (conversion, installation, hardware acquisition) ❑ Information (examples, other)	

(C2) Question/Topic		(C3) Notes/Answer
	 **STOP here until directed to continue to the next part.**	

Test Drive - Capture the Requirements

(C4) Restated problem(s) or request(s)	(C5)
Restated impact(s)	

NOTES

Test Drive 3

From an enhancement request for a data warehouse query application your group maintains. The enhancement request comes from the Southern Region Field Sales Support organization.

> **The system shall allow users at IBM-compatible PCs with 32 MB of memory to construct report templates without being connected to the network and the system shall work the same.**

Background

The system has two utilities for creating report templates: one is for on-line use (e.g., connected to the server); the other for off-line use (e.g., when a network connection is not available). Off-line operation is supposed to require a PC with at least 32 MB of memory. In reality, it requires a minimum of 64MB.

Attributes	Types	(C1) Notes/Issues
❑ Clear ❑ Correct ❑ Consistent ❑ Singular ❑ Testable, verifiable ❑ Feasible ❑ Design independent ❑ Traceable ❑ Clear	❑ Functional (user, mode) ❑ Exception handling (user, mode) ❑ Performance (quantitative) ❑ Design constraint (environment) ❑ Interface (interact with user, external components and other internal components) ❑ Communications (connectivity, access) ❑ Computer security and access ❑ Backup and recovery ❑ Implementation (conversion, installation, hardware acquisition) ❑ Information (examples, other)	

(C2) Question/Topic		(C3) Notes/Answer
		
	STOP here until directed to continue to the next part.	

Test Drive - Capture the Requirements

(C4) Restated problem(s) or request(s)	(C5)
Restated impact(s)	

NOTES

Test Drive 4

From a problem report for a data warehouse query application your group maintains. The problem report comes from the Southern Region Field Sales Support organization.

> **On the On-Hand Inventory screen, OHI-24, PARTS AVAILABLE BY PART NUMBER, the field for "Total Units Available at all Sites" is not updated so information is not available to personnel checking inventory at one location.**

Background

The field is supposed to be updated according to the latest design documentation. You do some further investigation with the software developers and confirm that it is updated following each monthly physical inventory.

Attributes	Types	(C1) Notes/Issues
❑ Clear ❑ Correct ❑ Consistent ❑ Singular ❑ Testable, verifiable ❑ Feasible ❑ Design independent ❑ Traceable ❑ Clear	❑ Functional (user, mode) ❑ Exception handling (user, mode) ❑ Performance (quantitative) ❑ Design constraint (environment) ❑ Interface (interact with user, external components and other internal components) ❑ Communications (connectivity, access) ❑ Computer security and access ❑ Backup and recovery ❑ Implementation (conversion, installation, hardware acquisition) ❑ Information (examples, other)	

(C2) Question/Topic		(C3) Notes/Answer
		
	STOP here until directed to continue to the next part.	

Test Drive - Capture the Requirements

(C4) Restated problem(s) or request(s)	(C5)
Restated impact(s)	

NOTES

Test Drive 5

From the statement of work for a warehouse fire detection system.

> **The system shall poll up to 128 sensors divided in up to 4 zones, with a maximum of 28 sensors per zone, at a rate of once every 30 seconds, except under high-load conditions, in which case the polling may by zone may take up to 60 seconds, or 90 seconds, if necessary, but an exception record will be logged on a journal printer (at the operator console) if the length of the polling cycle exceeds the above stated rate for more than 3 minutes and 10 seconds. After more than 4 minutes of extended polling, an audible alarm sounds.**

Attributes	Types	(C1) Notes/Issues
❑ Clear ❑ Correct ❑ Consistent ❑ Singular ❑ Testable, verifiable ❑ Feasible ❑ Design independent ❑ Traceable ❑ Clear	❑ Functional (user, mode) ❑ Exception handling (user, mode) ❑ Performance (quantitative) ❑ Design constraint (environment) ❑ Interface (interact with user, external components and other internal components) ❑ Communications (connectivity, access) ❑ Computer security and access ❑ Backup and recovery ❑ Implementation (conversion, installation, hardware acquisition) ❑ Information (examples, other)	

(C2) Question/Topic		(C3) Notes/Answer
		
	STOP here until directed to continue to the next part.	

Test Drive - Capture the Requirements

(C4) Restated problem(s) or request(s)	(C5)
Restated impact(s)	

NOTES

Test Drive 6

From a proposal approved as a top priority by Product Marketing (see note), to consolidate two systems your group maintains.

> **Develop SuperSystem, a single, integrated system for managing inventories at all repair depots and regional warehouses. Start with and use most of the Regional Warehouse Distribution Center System (DCS) on-hand inventory valuation and distribution capability. Incorporate repair-kit management functionality from the Depot Stock Management System (SMS). Develop real-time update and data replication functionality to replace current batch reporting under DCS and SMS. Develop interfaces to use bar code technology wherever applicable.**

We have to be able to offer this. Everyone is asking for it, the competition has it, and it should be easy to do through reuse and through standard data base tools.

The Boss.

Attributes	Types	(C1) Notes/Issues
❑ Clear ❑ Correct ❑ Consistent ❑ Singular ❑ Testable, verifiable ❑ Feasible ❑ Design independent ❑ Traceable ❑ Clear	❑ Functional (user, mode) ❑ Exception handling (user, mode) ❑ Performance (quantitative) ❑ Design constraint (environment) ❑ Interface (interact with user, external components and other internal components) ❑ Communications (connectivity, access) ❑ Computer security and access ❑ Backup and recovery ❑ Implementation (conversion, installation, hardware acquisition) ❑ Information (examples, other)	

(C2) Question/Topic		(C3) Notes/Answer
		
	STOP here until directed to continue to the next part.	

Test Drive - Capture the Requirements

(C4) Restated problem(s) or request(s)	(C5)
Restated impact(s)	

NOTES

Test Drive 7

From an e-mail from the CIO at a major customer, which has installed your comprehensive manufacturing plant management system at 14 sites world-wide.

> **All applications need guidance for Application Administrators to perform verification of database restore and procedures for recovery upon catastrophic system failure. Guidance should be in the form of a Manual that be referred to from the System Administrators Manual and contain proper restart, restore, and validation procedures to be used by Administrators after recovery of system prior to letting all users back on system.**

Attributes	Types	(C1) Notes/Issues
❑ Clear ❑ Correct ❑ Consistent ❑ Singular ❑ Testable, verifiable ❑ Feasible ❑ Design independent ❑ Traceable ❑ Clear	❑ Functional (user, mode) ❑ Exception handling (user, mode) ❑ Performance (quantitative) ❑ Design constraint (environment) ❑ Interface (interact with user, external components and other internal components) ❑ Communications (connectivity, access) ❑ Computer security and access ❑ Backup and recovery ❑ Implementation (conversion, installation, hardware acquisition) ❑ Information (examples, other)	

(C2) Question/Topic		(C3) Notes/Answer
		
	STOP here until directed to continue to the next part.	

Test Drive - Capture the Requirements

(C4) Restated problem(s) or request(s)	(C5)
Restated impact(s)	

NOTES

Test Drive 8

From the statement of work for which you are preparing a response.

> **Starting with and using the functionality of the San Corolla City System for tracking citations, develop a system which will consolidate all citations issued in Costanoa County by any City police department (in the County), or by the County sheriff. The citation should be automatically referred to the correct Municipal or County Court, which will be responsible for recording the disposition of the citation. Currently, data on unpaid fines is manually reported monthly by each Court Clerk to the State Department of Motor Vehicles so that drivers license renewals and vehicle registrations can be withheld until any outstanding items are resolved to the satisfaction of the Court with jurisdiction. The new system should automate this reporting.**

Background

You got a speeding ticket 7 months ago and you know what a nightmare it is waiting for the ticket to make its way through the different departments and courts so you could schedule traffic school. Your ticket was initially sent to the wrong Court, which delayed processing past the last date on which you were supposed to appear in Court of pay the fine. Everyone was very helpful and the dates were adjusted.

Attributes	Types	(C1) Notes/Issues
❑ Clear ❑ Correct ❑ Consistent ❑ Singular ❑ Testable, verifiable ❑ Feasible ❑ Design independent ❑ Traceable ❑ Clear	❑ Functional (user, mode) ❑ Exception handling (user, mode) ❑ Performance (quantitative) ❑ Design constraint (environment) ❑ Interface (interact with user, external components and other internal components) ❑ Communications (connectivity, access) ❑ Computer security and access ❑ Backup and recovery ❑ Implementation (conversion, installation, hardware acquisition) ❑ Information (examples, other)	

(C2) Question/Topic		(C3) Notes/Answer
		
	STOP here until directed to continue to the next part.	

Test Drive - Capture the Requirements

(C4) Restated problem(s) or request(s)	(C5)
Restated impact(s)	

NOTES

Test Drive 9

From a problem report submitted by the day-shift supervisor at a customer site.

> When I use the Web3M v6 to record a MACHINE DOWN problem with a piece of mountable equipment (e.g., sorter, stapler) which can be moved from one copying machine to another, the problem stays against the machine as well as the piece of mountable equipment. So I can't put the machine back on READY status so I can schedule work for it. Right now when I move the broken mountable equipment, I delete the original DOWN problem as a "misdiagnosis" and open a new problem against a fake machine I added to Web3M and which everyone knows to show as always DOWN. This really messus up our equipoment inventory and our maintenance numbers.

Background

Attributes	Types	(C1) Notes/Issues
❑ Clear ❑ Correct ❑ Consistent ❑ Singular ❑ Testable, verifiable ❑ Feasible ❑ Design independent ❑ Traceable ❑ Clear	❑ Functional (user, mode) ❑ Exception handling (user, mode) ❑ Performance (quantitative) ❑ Design constraint (environment) ❑ Interface (interact with user, external components and other internal components) ❑ Communications (connectivity, access) ❑ Computer security and access ❑ Backup and recovery ❑ Implementation (conversion, installation, hardware acquisition) ❑ Information (examples, other)	v6 is the latest version. This really happens. The system is working the way it is supposed to work. I heard from the sales support people that the work-around is driving the customer's internal financial auditors crazy because they keep looking for capital equipment and are told it doesn't exist. And the customer's Purchasing people are upset because they are supposed to complain to the contract Equipment Maintenance company if equipment stays down for more than 48 hours. And then when they beat the vendor up, they find out the equipment doesn't exist!

(C2) Question/Topic		(C3) Notes/Answer
		
	STOP here until directed to continue to the next part.	

Test Drive - Capture the Requirements

(C4) Restated problem(s) or request(s)	(C5)
Restated impact(s)	

NOTES

Test Drive 10

From a change proposal received from P. Blanchard at the BoardCom Travel Office for TAS, a Travel Accounting System your group maintains:

> **When a travel request is processed from the "pending transactions" screen, then saved and closed, the next travel request is not highlighted.**

Background

As the duty TAS expert, you quickly review the documentation to refresh your memory. The "pending transactions" screen lists outstanding travel-related transactions, including travel requests, expense reports, trip reports, and trip request approvals. The transactions can only be listed in order by origination date and time, so when any transaction is completed and closed, the transaction is removed from the list of "pending transactions", and TAS returns to the "pending transactions" screen, showing the first 24 outstanding, pending transactions, with the first transaction highlighted.

TAS Quick Reference

254 or 254	Move highlight up or down through the list of transactions 24 at a time	240 + 236	Highlight next transaction of the same type as the one that is currently highlighted
236 or 237	Highlight next or previous transaction in relation to the currently highlighted transaction	241 + 257	Highlight last (most recent) transaction of the same type as the one that is currently highlighted
252 or 257	Highlight the first or last transaction in the set of 24 currently displayed	241 + 252	Highlight first (the oldest) transaction of the same type as the one that is currently highlighted
242 + 252	Display the first 24 transactions; highlight first transaction (the oldest)	245	Begin processing currently highlighted transaction
242 + 257	Display the last 24 transactions; highlight last (most recent) transaction	241 + 182	Begin Verification (requires supervisor privileges)
240 + 237	Highlight previous transaction of the same type as the one that is currently highlighted	240 + *nnn*	Jump to transaction *nnn*

Attributes	Types	(C1) Notes/Issues
❑ Clear ❑ Correct ❑ Consistent ❑ Singular ❑ Testable, verifiable ❑ Feasible ❑ Design independent ❑ Traceable ❑ Clear	❑ Functional (user, mode) ❑ Exception handling (user, mode) ❑ Performance (quantitative) ❑ Design constraint (environment) ❑ Interface (interact with user, external components and other internal components) ❑ Communications (connectivity, access) ❑ Computer security and access ❑ Backup and recovery ❑ Implementation (conversion, installation, hardware acquisition) ❑ Information (examples, other)	

(C2) Question/Topic		(C3) Notes/Answer
		
	STOP here until directed to continue to the next part.	

Test Drive - Capture the Requirements

(C4) Restated problem(s) or request(s)	(C5)
Restated impact(s)	

Review Checklist

Attribute	Apply to Individual	Apply to Set	Definition	Problem Indicators	Examples	Corrective Action
Clear	X		Representatives of all affected parties should be able to read the requirement and come to a SINGLE, CONSISTENT INTERPRETATION	Imprecise, open-ended terms	"include, but not limited to", "support", “and/or”, “appropriate”, “maximize”, “minimize”, “consider”, “any”	Search for the "real" requirement and the additional information through interviews with the originator(s) or subject matter expert(s) (SME).
			Contains the information needed for all affected parties to begin work on THE NEXT STAGE IN THE LIFE CYCLE	Incorrect, complex, or overly precise sentence structure, bad grammar	“The operator chooses between A or B and C.”, “and”, “or”, “but”, “unless”, “above”, “below””, commas, semicolons	If appropriate, assist the SME through research, models, demos, use cases, or prototypes.
				Ambiguous, multiple meanings possible	“it”, “this”, “that”, “which”, “above”, “below”, “previous”, “following”, “next”	Validate the requirement with the originator or SME.
Testable, verifiable	X		Description is sufficient that testers can see where they could devise the means to determine whether the requirement was properly implemented in the product	Open-ended, imprecise, ambiguous terms		
Design independent	X		Describes “what”, not “how”	Operation (not function)	Press the F8 key Add a database field, so ...	
Correct	X		Accurate (and complete) as determined by the customer and user or their representative(s)	Something missed or misinterpreted		Determine if this is the "right" requirement. Validate the restated requirement with the originator/user or their representative(s). Surface for a decision/solution.
Traceable	X		Able to link the requirement to a specific origin	Cannot determine where or who	Federal Regulations, lots of customers, the competition	Pinpoint the source. If appropriate, compare it to its source to determine if the stated requirement meets the intent of the source
Complete	(x)	X	Internal references are all resolved. No referenced or implied elements or activities are missing All attributes of *identification* that can be determined at this time are supplied.	Something missing		Track down the missing information; query the originator.
Consistent	(x)	X	Not in conflict with other requirements (both technical or non-technical, policies, regulations, laws). Do not duplicate or overlap			Resolve the conflict by determining relative priority or through negotiation with affected stakeholders. Change or eliminate conflicting or overlapping requirements.
Feasible	(x)	X	It is possible to implement the requirement set within the known capabilities and limitations of the system and its environment: technically, and within costs and schedule constraints			Refer, defer, identify trade-offs (schedule, performance - What can we do?), decline, play chicken.

Attributes of a Requirements Process

Baselined - set	Requirements are **baselined**. Once a set of requirements have been reviewed and approved, changes are made only through change control procedures.
Versioned - individual and set	Requirements and requirements sets are **versioned**. At any point in the process, the current version of a requirement or set of requirements can be determined.
Identification	Requirements templates and review checklists ensure that all available information is provided.
Captured	Requirements and requirements sets are **captured**. They are recorded in a form which can be baselined, versioned, distributed to all impacted parties, stored, retrieved, and maintained. There is no restriction on medium (e.g., paper, web) or notation (e.g., text, diagrams).
Visible	Requirements, requirements sets, and changes to requirements and requirements sets are **visible** to all impacted parties.
Responsibility and authority - approval and change	**Responsibility and authority** for requirements, requirements sets, and changes to requirements and requirements sets are clearly defined for all phases of the life cycle (e.g., from initial receipt as an incident to the release of a product that satisfies the requirements). Individuals have the skills, knowledge, and time they require to perform their requirements-related tasks. There may be a change control board (**CCB**) which is responsible for the initial approval of requirements and requirements sets and for the on-going approval of changes to requirements and requirements sets. There may be multiple boards with complementary responsibilities. Different boards and/or board members may be active at different points in the life cycle.
Internal peer review; external review	Requirements and requirements sets and changes to requirements and requirements sets are **peer reviewed** by all impacted parties within the organization (e.g., hardware, software, system test, publications, operations, support) before they are reviewed by external parties (e.g., customers).
All types of requirements addressed	The process addresses **all** requirements – including supplied and derived requirements, functional and technical requirements, and non-technical requirements (e.g., schedule, budget).
Requirements specification - package of requirements, interactions, dependencies, etc.	Requirements are considered both as individual items (e.g., as a Change Request or Problem Report) and as they interact in a set of requirements intended for a release (e.g., in a **requirements specification** for a software release or project).

XtremelyAgile Scenario

Customer (C): We're looking at a low-impact, user-friendly way we can standardize company-wide administrative functions and improve day-to-day communication between the various departments.

Engineer (E): So, it sounds like you want a Windows application?

C: Well, yes. We have some of those already and people are comfortable with them. When they have the training.

E: Tell me more about how you envision this standardization working.

C: Well, we have a large number of functions, e-mail, inventory record keeping, performance evaluation, contingency planning, facilities maintenance, and preventive maintenance, that we want to tie together – and we will have more coming on as the organization diversifies – including configuration management, document control, and a link to Corporate Central IS Services. Our functions are all over the map geographically – including those that are used by employees traveling anywhere in the world. Which reminds me, we'll also need Voice over IP for telephone communication over the Internet at the lowest possible rates from anywhere in the world.

E: It sounds like we'll need to take an object-oriented approach hanging discrete application modules off a central data base which can be instantiated in multiple, synchronized servers for maximum availability of mission critical functions. We'll also need to invoke CORBA and SOAP for access to other data bases and existing applications. And we'll need to develop or integrate wireless peripherals.

C: That sounds reasonable, I suppose. Have you done something like this before?

E: Of course. Many times. Tell me what type of budgets and time frames we're looking at.

C: Well, we have $200,000 in the current fiscal year budget that we have to burn and we need to be operational with the first components in less than 6 months.

E: Can't be done.

VIGNETTES

Indicate any CMM, management, or operational issues relevant to each brief scenario.

V1	Each accepted requirement is assigned a tracking number, which is included in the change section of every derived document and in affected code modules as a comment. A requirement remains open until the version of the product in which it is implemented is released for general customer availability. Weekly metrics on status and on-time completion are maintained.
V2	The Engineering Change System (ECS) is used for managing requirements changes, engineering-generated technical changes, and incident reports that have been verified in our support laboratory as real problems.

V3	BSC's software applications are licensed to customers for installation and operation in their own data processing centers. Our applications are also used by a division of BSC that outsources data processing for customers who do not have sufficient data processing capacity. Because of the real-time nature of our applications, the BSC outsourcing center has a small production support software engineering group that handles installation of software upgrades and that makes emergency patches to the software.
V4	As part of BSC's continuous improvement and problem prevention initiative, if there is no impact on development schedules, software engineers are encouraged to improve code that they encounter as part of implementing their assigned, approved change requests. This ranges from cleaning up formatting inconsistencies to improving functionality and capacity to prevent problems that have not yet been reported.

V5	BSC's core product components serve as the foundation for customer-specific customization projects. These projects are run by customer project managers who act as business unit managers with profit and loss responsibility for their projects. As a result, these project managers work with their customer to identify ways to grow their projects.
V6	All projects will adapt or adopt Requirements Engineering Complete (ReCo), the requirements management tool developed by the director of software engineering at his last company.

The Requirements Specification (SRS)

The following is divided into several sections and subsections:

Section	Title	Summary
A	An outline for a generic SRS	The headings for a template for a generic SRS
B	About the Content of an SRS	Suggested conventions and standards to follow in writing and reviewing SRSs.
B.1	Assessing the quality of a requirement	
B.2	The attributes of identification	
C	Typical content by section	Suggestions, and examples for each section in the outline.
C.1	Example 1: By function	An example of requirements organized by function
C.2	Example 2: By mode	An example of requirements organized by mode
C.3	Example 3: By user	An example of requirements organized by user

A. Outline for a Generic SRS

The following is a basic outline for a generic SRS. As a model, it is not usable "as is" for anyone. It omits sections that are critical for some organizations; it contains other sections that are irrelevant for some organizations. It is intended to provide useful ideas, suggestions, and starting point for individuals defining SRSs for their organizations, technologies, and applications.

The template is structured so that, if the size of a project warrants it, it can be written and reviewed in two increments: Sections 1 and 2, to establish a verified framework, and then Section 3, detailed requirements.

In all cases, if a section or block of content is found in all SRSs produced by the organization (e.g., a Human Computer Interface (HCI) standard), provide that information in a separate document and – if worthwhile - refer to it in the SRS. Include only a brief reference to the standard in each SRS, to record, at least, the version current at the time the SRS is being written. Such a reference is particularly useful when there is a possibility that the work may be outsourced.

1. Introduction
    - 1.1 Purpose
    - 1.2 Scope
    - 1.3 Definitions, acronyms, and abbreviations
    - 1.4 References
2. Product description
    - 2.1 Product roadmap
    - 2.2 Target environment
        - 2.2.1 Interfaces
        - 2.2.1.1 Users
        - 2.2.1.2 Hardware
        - 2.2.1.3 Software
        - 2.2.1.2 Communications
        - 2.2.2 Operating modes
        - 2.2.3 Delivery and setup
    - 2.3 Functions
    - 2.4 Users
    - 2.5 Constraints
    - 2.6 Assumptions and dependencies
3. Requirements

Appendixes

B. About the content of an SRS

While there is a section titled *Requirements* in the SRS, it contains detailed descriptions of requirements that complement requirements statements that appear in other sections of the SRS. All of the statements (requirements and others) contained in the SRS follow the same stylistic conventions. Recommended conventions are:

- *Shall* denotes a requirement for the product. For example, *The system shall produce reports that comply with Version 2 of the Federal Standards for Cost Accounting.*
- *Should* denotes an objective for the product. For example, *The system should perform multi-location inventory searches at least 10% faster than any competitive product – from starting the search (after the search criteria are entered) to the presentation of the list of locations*. The objective allows the SRS to capture "customer" objectives that cannot be realistically quantified or expressed as requirements without running the risk of over- or under-specification. Success in meeting objectives is monitored (but not necessarily managed) and is the subject of on-going discussion with the customer as the design and implementation progress. An objective typically supplements or stretches a "shall" statement that establishes a lower bound for the objective. For example, an SRS could include both the objective *10% faster* and the general requirement that *System response times for identical queries shall not exceed those of the current system by more than 2%*.
- *Is*, *are*, *will*, and *present tense of verbs* are statements of fact or intention. For example, the following are statements of fact:
  - *The user is able to read and understand English.*
  - *The user reads and understands English*.

B.1 Assessing the quality of a requirement

All of the statements contained in the SRS are systematically reviewed by all relevant stakeholders to ensure that each individual statement and the complete set of statements in the final, approved SRS are:

- Clear and unambiguous
  Representatives of all affected parties read the statement and come to a single, consistent interpretation. There is consistent, agreed-upon use of stylistic conventions
- Testable, verifiable
  The statement is specific enough that developers and testers agree that they could devise the means to determine or measure whether the statement is properly implemented in the product.
- Design independent
  The statement describes "what", not "how". It describes a need or a problem - not a solution. It describes function, not operation. The following example is based on an enhancement request for an existing system. In the current version of the system, when a stock check at one depot shows zero on hand, the clerk has to waste time changing screens, entering depot codes one by one, and reentering part numbers to find out if any other depot has on-hand inventory of the part. Then the clerk has to go to the screen to request an inter-depot transfer and enter all of the information about the two depots and renter all of the data about the part.

This	Not this
The system shall allow the inventory clerk to go directly from the Stock-on-Hand screen to identifying which depots have inventory available (without requiring that the operator enter a specific depot number and search depot by depot), to requesting an inter-depot transfer (without having to reenter any data about the part.).	Fields on the Stock-on-Hand screen shall show the on-hand inventory of the part at all locations. A field below the list of depots and on-hand inventory amounts shall allow the clerk to enter a depot number. A button on the screen beside the field in which the clerk enters the depot number shall allow the stock clerk to jump directly to the "Request Inter-Depot Transfer" screen.
Other than requiring that the operator supply the number of the depot from which he or she is requesting the transfer, the system shall automatically capture all of the static location, cost code, etc. information for the transfer request.	The system shall fill in all of the static data and carry forward the part information automatically on the "Request Inter-Depot Transfer" screen.

It should be noted that the background information that preceded the "This/Not This" table is also valuable for inclusion in the SRS as a statement of fact, identified as information.

- Correct
  The statement is accurate (and complete) as determined by the customer and user or their representative(s) (e.g., Marketing)
- Traceable
  The statement is linked to a specific origin – a predecessor document included in *1.3 References.*
- Complete
  All appropriate attributes of identification (see *the attributes of identification*, below) are provided for each requirement. All internal references or implications are resolved (e.g., Does requirement x have any security implications that might lead to an additional requirement).
- Consistent
  The statement is not in conflict with other requirements.
- Feasible
  It is possible to implement the requirement set within the known capabilities and limitations of the system and its environment – both technically, and within and established cost and schedule constraints.

B.2 The attributes of identification

Each individual statement in the SRS has a unique identifier which is used to track the status of the requirement statements through each phase of the development process. In addition, the organization uses the SRS or an associated database to capture and maintain, as the project progresses, the following information for each statement in the SRS:

	Requirement (shall)	Objective (should)	Fact (is, …)
- Type	X	X	X
- Necessity	X	X	
- Stability	X	X	X
- Allocation	X		
- Parent (requirement)	X		
- Source (document, paragraph)	X	X	X
- Rationale	X	(X)	
- Verification (method, documents)	X		
- Changes	X	X	
- Current status	X	X	
- Planned or applicable release	X		

For requirements statements, the *shalls*, the following types typically apply:

- Program (process, legal, schedule)

Technical

- Functional (user, mode)
- Exception handling (user, mode)
- Performance (quantitative)
- Design constraint (environment)
- Interface (interact with user, external components and other internal components)
- Communications (connectivity, access)
- Computer security and access
- Backup and recovery
- Implementation (conversion, installation, hardware acquisition)
- Information (examples, other)

C Typical content by section

		Section	Contents, Description, Examples, and Comments
1		**Introduction**	The *Introduction* provides the background for the product or project to which it applies. The content of the *Introduction* is all in the sub-sections.
	1.1	**Scope**	Start with "This document applies to:" and then identify the product(s) and project(s) to which the document applies. For example, *This document applies to:* - *Release 5.2 of OPTI-2000* Or, - *To the Web-link Embedding Project* Or, - *To the Speech Recognition Input and Output Engine Project developed under Contract 02-0004-05 for United Terrestrial LLC"* NOTE: This section contain only statements of fact.
	1.2	**Definitions, acronyms, and abbreviations**	Define any specialized or potentially ambiguous terms used in the SRS, and any acronyms or abbreviations used in the SRS. Consider, at least: - Reviewers, e.g., Marketing, the customer, Systems Engineering, - Designers and architects, who use the SRS as input for their activities, and - Other SRS authors who will use the SRS as a baseline for their follow-on work or who are using the SRS to coordinate the development of their SRSs. NOTE: This section contain only statements of fact.
	1.3	**References**	List all of the documents referred to in the SRS. For each document, provide the title, control number, author, source or location, and current version. Based on the structure of the source document, consider whether to include a section number in the reference. Sub-section 1.3 of the SRS becomes the basis for determining the impact of changes. The information in this sub-section is frequently placed at the end of the SRS. NOTE: This section contain only statements of fact.
2		**Product description**	This section provides a framework for the detailed requirements in Section 3. The content of the *Product Description* is all in the sub-sections.
	2.1	**Product roadmap**	Describe, as appropriate, how this product fits into the organization's product mix. For example, *OPTI-LT* [the product being specified] *is a low-end, entry-level system for users who cannot afford or justify the purchase of an OPTI-2000 system and the associated hardware*. Alternatively, describe how this version of the product fits into the overall strategy for the product. For example, *Release 5 currency conversion functionality for North American banks, which is planned to be enhanced in Release 7, to address the needs of multi-national banks.* If this is an independent product (e.g., developed under a contract for a specific customer), state this. NOTE: This section contain only statements of fact.
	2.2	**Target environment**	Characterize all aspects of the environment in which the product will be used. Describe, as appropriate, the rules governing the interaction between this product and external entities. Block diagram(s) are particularly appropriate to support this description. This sub-section is frequently divided into categories (e.g., *Interfaces* and *Operational Modes*), which are further decomposed into related topics (e.g., *Interfaces* is decomposed into *Users*, *Hardware*, *Software*, and *Communications).* The content of the *Target Environment* is all in the sub-sections.
	2.2.1	**Interfaces**	This section lists external entities with which the product interacts directly. The nature of the interaction is defined in *2.3 Functions.* The content of the *Interfaces* is all in the sub-sections.

	Section	Contents, Description, Examples, and Comments
2.2.1.1	**Users**	Define or reference any applicable Human Computer Interface standards or related rules that must be supported by the product being developed. For example: - Established use of a particular function key for a common task - Windows Developers Tool Kit Standard Human Computer Interface Specification v 21.6
2.2.1.2	**Hardware**	List any hardware components (other than communications) with which the product must interact directly; define or reference any applicable protocols that must be supported by the product. For example: - A 12-port DigiDog board supporting bi-directional S-BUS message protocols from 155X-series SmartSensors - Use Windows QT printer and file management services
2.2.1.3	**Software**	List any software products (other than communications) with which the product must interact; define or reference any applicable formats, etc. that must be supported by the product. For example: - Use standard Windows file management and print services - Access the Corporate Customer Data Base using SQL - Be fully compatible with the Norden Disaster Recovery (NDR) utilities and routines implemented by MIS (at client, server, and mainframe level)
2.2.1.4	**Communica-tions**	List any communications interfaces (e.g., LAN, WAN, etc.); define or reference any applicable protocols, etc. that must be supported by the product.

	Section	Contents, Description, Examples, and Comments

2.2.2	**Operating modes**	Characterize the distinct modes in which the target user organization operates – as they relate to the product being developed. Depending on the scope of the product being specified, modes may be associated with organizational units or locations. The functions that occur in each mode are described in *2.3 Functions*.
		
		For example, describing an organization for which a bug-tracking application is being specified:
		
		- *Customer Service and Support (CS&S)* is divided into two distinct organizations:
		  - *Support Center*. Customer calls are accepted by two tiers of technicians from 05:00 to 08:00 Monday through Saturday PST.
		  - *Advanced Engineering*. Escalated (e.g., severe or high priority) customer calls are addressed on a 7x24 basis. They currently plan for the system to be unavailable from 01:00 to 02:30 PST Monday through Friday for minor maintenance (e.g., incremental database backup, hardware replacement) and 01:00 to 06:00 PST Saturday for major maintenance (e.g., full backup, software upgrade, server maintenance)
		
		A more effective alternative to the *Customer Service and Support (CS&S)* example immediately above would be:
		
		- *Customer Service and Support (CS&S)* is divided into two distinct organizations: *Support Centre (SC)*, which handles customer calls and *Advanced Engineering (AE)*, which handles escalated, high priority customer calls and which creates patches. These two organizations operate in three overlapping modes:
		  - *Mode Active-Full:* Customer calls are accepted by two tiers of SC technicians from 05:00 to 08:00 Monday through Saturday PST.
		  - *Mode Active-Advanced:* Escalated (e.g., severe or high priority) customer calls are addressed by AE personnel on a 7x24 basis.
		  - *Mode Active-Advanced-Offline:* The current system is unavailable from 01:00 to 02:30 PST Monday through Friday for minor maintenance (e.g., incremental database backup, hardware replacement) and 01:00 to 06:00 PST Saturday for major maintenance (e.g., full backup, software upgrade, server maintenance)
		
		For example, describing the modes of operation for an organization for which a credit-checking application is being specified:
		
		- *Finance* operates four regional centres. The centres are in NCSA (North, Central, South America; Dallas USA), EMEA (Europe, Middle East, Africa; Bonn GER), AP (Asia Pacific; Adelaide AUS) and China (Jinan CH)
		- Mode OPEN: Each centre is fully staffed and operates from 07:00 to 17:00 local time Monday through Friday.
		- Mode LOCKED: Mandatory database backup is performed by the MIS department from 18:00 to 19:30 local time Monday through Thursday and from 18:00 to 21:00 on Friday. During these times *systems are not available – i.e., the centre cannot function during these periods, and other centres cannot access data from the centre.*
		- Mode CLOSED: At all other times each center is nominally closed (e.g., individuals may be working).

2.2.3	**Delivery and setup**	Describe how the product will be distributed, delivered, and setup. For example, they product being specified may be intended for delivery on CD-ROM or by download: - For automated installation over existing versions (e.g., upgrade) or on a clean system. - For installation over a competing system, requiring data conversion. - For customization by a trained value-added reseller

	Section	Contents, Description, Examples, and Comments
2.3	**Functions**	Summarize the major functions that the product being specified is expected to provide. Once again, a graphical representation can be particularly useful in constructing this summary. This section contains initial, high-level decompositions of the functions. This section categorizes *functions* – it is not a design. To avoid creating a huge document, consider the various audiences for the SRS. Describe additional, detailed decomposition, which is of interest to narrower, more specialized audiences, in separate documents, which are referenced in this section of the "master" SRS. It is also particularly useful to identify the modes (from *2.2.2 Operating modes*) with which each function is associated. For example, in specifying a financial application, *2.3 Functions* states that: The application shall provide functionality in the following areas: customer account maintenance, billing, and processing payments. [Note that In *2.2.1.3 Software interfaces*, the interfaces to the Order Processing, Shipping, and Credit Checking systems were defined.] *2.3 Functions* is then broken down into subsections in which each of the major functions is further decomposed (e.g., subsections for *Customer account maintenance*, *Billing*, and *Processing payments*. The subsection for *Customer account maintenance (CAM)* specifies that CAM addresses: - Credit verification - Customer data maintenance (address, contact, etc.) - Customer status maintenance (ok, hold temporary, hold permanent) - Order summary reports (3 months, 6 months, 12 months) - Aging (monitoring, notification, reporting) - Payment history report (3 months, 6 months, 12 months) Corresponding subsections describe *Billing* and *Processing payments*.
2.4	**Users**	Characterize the individuals involved in performing the various functions listed in *2.3 Functions*. This section of the SRS is typically divided into sub-sections for each class of user. Useful attributes of these individuals include: - Functions performed (as listed in *2.3 Functions*) - Level of education - Level of training on the application being specified - Professional certification - Level of experience on computers, on similar applications, in the application domain - Cultural issues (e.g., with respect to Collections) - Language limitations

Section	Contents, Description, Examples, and Comments
2.5 Constraints	From one perspective, an SRS contains nothing but constraints. The SRS focuses developers on solutions that address specific needs. The constraints described in *2.5 Constraints* are those that are related to technical decisions, decisions that are normally left to the engineering team (e.g., choice of a development platform, a data base vendor, a configuration management tool). As part of the documentation and review processes, each of these constraints is questioned to ensure that it has a basis in the customer needs. For example, - The product shall run on a WINTEL platform with a maximum of 64MB of memory. *[Questioning reveals that the customer has thousands of these systems deployed and has no intention of upgrading them.]* - The development group shall use CodeKeeper for all a configuration management. *[Questioning reveals that the customer uses CodeKeeper for internally developed software and drawings and requires that all development subcontractors use the same system to facilitate the exchange of source code and other project artifacts.]* - The software shall run under Windows 98. *[Questioning reveals that 80% of the customers in the target market have Windows 98 installed and cannot justify the expense of upgrading to Windows 2000. The other 20% have a combination of Windows 95, ME, 2000, and XP. It is highly uncertain as to when or if the Windows 98 users will upgrade; and, if they do, it is not clear whether they will move to 2000 or XP. Many users are so determined not to upgrade, they are buying systems with XP installed and replacing it with Windows 98.]* In each case, the elaboration obtained through questioning is added to the description of the constraint (1) to reinforce the validity of the constraint and (2) to minimize the number of times the reason for the constraint has to be independently rediscovered. The relationship between *2.5 Constraints* and 2.*2. Target environment* is potentially confusing. The confusion typically results in repetition within an SRS (e.g., "I'm not sure where it goes, so I'll put it in both places") and inconsistency between SRSs (e.g., one author defines hardware constraints as part of the interface, in *2.2.1.2 Hardware [interfaces]*; another author defines them in *2.5 Constraints*). There is a logical distinction that can be conveyed through training and perfected through exercises and writing specification. *2.2 Target environment* focuses on the interactions between the product and other entities. Specific limitations and the associated justifications imposed by the technical entities – hardware and software – are documented in *2.5 Constraints*.

	Section	Contents, Description, Examples, and Comments
2.6	**Assumptions and depend-encies**	In some cases, the most effective and efficient way to verify detailed assumptions associated with statements made in any of the other sections of the SRS is to document the assumption (flagged for reviewers as "ASSUMPTION") in line with those statements. For example, if no user input has been received regarding language support and localization, the SRS could include the following in *2.2.1.1 Users* : *ASSUMPTION: All users, including Tier 1 personnel, are able to read English.* Or, *ASSUMPTION: All users receive at least two hours of training prior to operating the system.* Once an assumptions is verified, the assumption is rewritten as a factual statement, incorporating any feedback from reviewers and approvers. In *2.6 Assumptions and dependencies,* list assumptions pertaining to the content and functionality of the product that cannot be verified because they pertain to future events that are outside the control of the development organization or the customer and which put success at risk. For example, - If the customer specifies that the product integrate or incorporate future releases of 3rd party hardware or software products, interfaces to these items are called out in the appropriate sections of *2.2 Target environment.* These items are also listed in *2.6 Assumptions and dependencies*. There may be a forward reference in *2.2* to *2.6*. - Similarly, if this product relies on the work products of another internal project or group, those products appear in *2.2* and *2.6*. - If the functions of a mathematical model are described in a clear and unambiguous manner, but if there is serious concern about the ability to create such a model, that concern is recorded in *2.6 Assumptions and dependencies.* There is another class of assumption, related to those cases in which the development team has concerns about the stability of agreed-upon, well-defined needs or requirements. For example, even if the customer states and confirms in response to a direct question that *all users are able to read English*, the development team may anticipate that at some point in the development process, the customer will request support for multiple languages. By documenting this assumption, the team may be able to create a design that addresses the immediate need without any unnecessary cost to the customer or the organization, but which does not require any rework to add support for additional languages. The most effective and efficient way to record these assumptions is through the stability rating and elaboration for the requirement in *3 Requirements*.
3	**Requirements**	*3. Requirements* begins with a description of how the requirements are organized into subsections. The subsections of *3* contain detailed requirements that supplement those in the previous sections of the SRS. The level of detail is correct when the information is: - Sufficient for designers to design products that the customer perceives as satisfying the requirements. - Not so voluminous that customer representatives are unable to confirm that their needs have been recorded correctly. It is the responsibility of the SRS authors, the systems designers and the customer representatives to determine whether this balance has been achieved through reviews of the SRS. Typically, as a design evolves, the designers raise additional questions and uncover unanticipated options that require revisions and re-reviews of the SRS. In addition, especially for longer-duration projects, customer-originated changes require changes to the SRS. The most effective means of addressing the evolving requirements is through short projects that define increments of functionality and through close, on-going cooperation and frequent, but controlled, contact between the customer representative(s)and the development team. In addition to the quality of the content, as described above in *About the content of an SRS*, the organization of the requirements statements contributes significantly to its effectiveness in communicating with reviewers and designers. ***Organizing requirements*** The organization of the earlier sections of the SRS identifies dimensions or variables available for organizing the detailed requirements: modes of operation, functions, and classes of users. While it is sometimes difficult to maintain design independence, the organization of the

	Section	Contents, Description, Examples, and Comments
		requirements in no way reflects the design of the product or its components. Based on past experience, as the detailed requirements are constructed and verified, the earlier sections of the SRS will require modification. In particular, the process of creating the detailed requirements may lead to changes in the way other sections are organized, including *2.2.2 Operating modes*, *2.3 Functions.* Defining an initial organization for this section begins by selecting one dimensions as dominant: ▪ If the functions remain relatively stable within modes, the initial draft of the SRS may start with functions, describing any mode-based differences by function, and identifying user interactions within each function. ▪ If the system operation varies significantly from mode to mode, the initial draft of the SRS may group functions within modes, and identify user interactions within functions. ▪ If the system operation is driven by the different organizations, the initial draft of the SRS may group functions by users, and identify mode-based differences within the functions. Samples of requirements organized around each of these dominant dimensions are found in Sections C.1 through C.3, below.
	Appendixes	Supplementary material as appropriate.

C.1 Example 1: By function

This example presents an extract from a working draft of the requirements for R-SUPPLY, a new automated system for ordering computer supplies from AllSorts.

Function is the primary dimension for organizing the requirements. Users' responsibilities and the few variations based on mode are described in the context of function.

C.1.1 Background from the first two sections of the SRS

All of the following information is provided in first two sections of the SRS.

Authorized customers, who have little or no training or experience, currently order computer supplies by telephone. The customer finds the items in the current AllSorts paper catalogue, and calls an order taker at AllSorts who enters the order data for the customer. There are currently 8 order takers who work 06:00 to 19:00 PST. The customer has an AllCard, a "credit card", complete with encoded magnetic stripe, which is only good for ordering supplies from AllSorts. Customers get as many AllCards as they want – the security of the AllCards is the customer's responsibility. AllCards are issued by the First International Bank (FIB), which handles all cancellations, billing, aging, collections, and fraud investigations. AllSorts and FIB Information Systems are fully integrated:

- AllSorts reports AllCard transactions, charges and credits to FIB as if they were credit card transactions.
- Authorized AllSorts personnel have all required access to current account balances and aging by AllCard and by customer.

92% of the customers have PCs with Windows. Of those customers who have Windows PCs, 8% are running Windows 3.1; 28%, Windows 95; 52%, Windows 98 SR2; 22%, Windows 2000.

8% of the customers do not have PCs with Windows. Of those customers who do not have Windows PCs, 80% have a variety of Apple computers, 16% have an Intel platform with a version of Unix, 4% have an Intel platform with DOS. Actually, 2 customers have machines running CP/M.

C.1.2 The requirements

3.0 Requirements

This section of the SRS is organized by the major functions of R-SUPPLY.

3.1 General – for all of R-SUPPLY

R-SUPPLY shall:

- Allow customer personnel to access only those functions for which they are authorized.
- Allow AllCard access authorization verification at any workstation
- Require minimal training for successful operation.

- Operate 24x7x365.
- Allow authorized AllSorts personnel to perform all functions on behalf of authorized customer personnel, where authorization can be confirmed over the telephone.
- Allow 20 orders to be placed simultaneously and TBD order status checks and TBD account status checks.

R-SUPPLY should:

- Use all possible services of the Internet both to minimize costs and to ensure that AllSorts appears to its customers and institutional investors to be in tune with the current trends in technology.
- Mimic, as closely as possible, the look and feel of the current system (e.g., order forms, catalogue format) to minimize retraining.

3.2 Place an order

R-SUPPLY shall:

- Allow authorized customer personnel to order computer supplies from AllSorts without requiring the intervention of an AllSorts order taker. Authorization is provided based on the AllCard.
- Ensure that the AllCard is valid prior to accepting an order.
- If the supplied AllCard is not valid, halt processing and notify the 24-hour FIB customer service and fraud elimination staff
- Derive all delivery and accounting administration information from the AllCard number.
- Accept a customer-supplied part number and quantity.
- Respond to the customer with the part description unit price, a calculated extended price, an indicator of whether the item is on back order, and the current total of the order, excluding taxes, shipping, and handling.
- Allow the customer to
  - Start a new line item
  - Change the part number or quantity and continue working with the current line item
  - Delete the line item completely and start over
  - Delete the line item and quit
  - Review the current order and make any appropriate changes at any time during the order placement process
  - Allow the customer to authorize partial shipments
  - Allow the customer to specify a shipment method for the whole order or specify different shipment methods by line item

    …
- Present the total cost of each type of shipping and handling specified.
- Recalculate the order total to include the total for all types of shipping and handling specified.

  …

3.3 Order status check

R-SUPPLY shall:

- Allow authorized customer personnel to check on order status without requiring the intervention of an AllSorts order taker.
- ASSUMPTION: Anyone who can place an order can check on the status of any order placed using that AllCard.
- ASSUMPTION: A class of users needs authorization to check on any order placed for any AllCard issued to the customer.

  …

3.4 Account status check

R-SUPPLY shall:

- Allow authorized customer personnel to check on account status without requiring the intervention of an AllSorts Customer Account Service Representative.
- ASSUMPTION: Contain or have access to a list of authorized customer personnel and that some sort of password can be provided to those selected customers.

  …

C.2 Example 2: By user

This example presents an extract from a working draft of the requirements for a PPERS, an automated product problem reporting system for use at SoftCam.

User is selected as the primary dimension for organizing the requirements. Functions and the few variations based on mode are described in the context of users.

C.2.1 Background from the first two sections of the SRS

All of the following information is provided in first two sections of the SRS.

SoftCam's first product is currently completing beta test and nearing release. The product addresses the needs of a small, highly technical, niche market. SoftCam's product is a robust, low-cost, fully functional alternative to the mature products offered by its established competitors. These competitive products have not aged well and have become increasingly more difficult to use and to maintain.

In addition, SoftCam's few competitors have high overhead costs and they have been treating their customers like cash cows for the last five years. Based on the beta test results, the product is virtually bug free and SoftCam anticipates a relatively low volume of calls.

After extensive research, SoftCam management and the Senior Technical Committee decided to make rather than buy a problem reporting system. The principle reasons are because the unique circumstances of SoftCam's business make unnecessary most features of any of the commercial systems considered. SoftCam's customers have a high level of technical proficiency, and SoftCam's current business model calls for Engineers to handle customer calls. As part of the learning process, junior engineers will perform the initial screening and handle reports that involve: operator errors, previously resolved problems, and previously reported problems that are still being resolved. Senior engineers will handle those

While management recognizes that this support strategy will change, it will not happen for at least three years. At that time, a transition to a commercial product will be reconsidered.

C.2.2 The requirements

3.0 Requirements

3.1 General

PPERS should incorporate off-the-shelf tools whenever possible.

3.2 Customers

PPERS shall:

- Accept problems reports from customers.
- Provide a form with required and optional fields to ensure that contact information, product information, and problem information are gathered. The content of the form is TBD.
- Validate forms fields as they are entered: valid customer contact information, valid product and configuration information. Other validation TBD.
- Allow customers to enter a severity factor
- Assign a unique identifier and date and time stamp to the submitted form.
- Provide customers with a hard or soft copy of the completed form, with the unique identifier

3.3 Junior engineers

PPERS shall:

- Calculate a priority based on the input information
- Provide a list of prioritized reports which includes aging information
- Allow junior engineers to self-assign reports
- Allow junior engineers to record research and resolution information
- Allow junior engineers to escalate reports at any time
- Automatically escalate HIGH PRIORITY reports after 48 hours.
- Automatically assign HIGH SEVERITY reports after 24 hours.
- Automatically escalate any report after 5 days
- Allow junior engineers to CLOSE all except HIGH PRIORITY and HIGH SEVERITY reports

- Require an engineer's authorization to close HIGH PRIORITY and HIGH SEVERITY reports

…

3.4 Engineers

PPERS shall:

- Provide a list of prioritized, escalated reports which includes aging information
- Allow engineers to self-assign reports
- Allow engineers to record research and resolution information
- Allow engineers to close reports

…

3.4 Development manager

PPERS shall:

- Provide graphical summary and detailed reports on report trends, aging, by customer, by cause (product component), and by time to close.
- Include a query capability that allows the manager to create custom reports from the repository.

…

C.3 Example 3: By mode of operation

This example presents an extract from a working draft of the requirements for a Customer Support System (CSS) at SoftWest. The requirements specification will be the basis for soliciting proposals and for determining whether to make or buy the system.

Mode of operation is selected as the primary dimension for organizing the requirements. Functions and the roles and responsibilities of users are described in the context of users.

C.3.1 Background from the first two sections of the SRS

SoftWest develops software products which it sells as stand-alone products and pre-installed on Dahl ruggedized lap top computers running Windows QT. SoftWest is a Dahl distributor and preconfigures and tests each lap top before it is shipped. The SoftWest Technical Support organization currently offers 12 hours of free call-in or e-mail service for its software for a period of one year from the initial call. These limitations are not enforced or enforceable. Few customers exceed the 12-month/12-hour restriction, and those that do include the most committed customers, who have the highest referral value.

During the 3-month warranty period, hardware service is provided directly by Dahl through its network of licensed service providers. After 3 months, Dahl offers an extended contract with options for mail-in (to regional service centers) and for carry-in (to participating, licensed service providers). Dahl is currently planning a third option, for on-site service.

Based on input from its User Advisory Board, SoftWest is planning to supplement its 12/12 warranty service with 3 additional tiers of service, incorporating support and software maintenance.

- **Bronze Service** – offered at the time the software or system is sold as a one-time, non-renewable contract – unlimited telephone support for 1 month from the first call; unlimited e-mail support for 5 months from the end of the telephone support period; unlimited access to the web based knowledge base; access to download service packs (bug fixes) for 3 months from the first download.
- **Silver Service** – offered at any time under a one-year renewable contract – 24 hours of telephone support and unlimited e-mail support; unlimited access to the web based knowledge base; access to all service releases (bug fixes)
- **Gold Service** – offered at any time under a one-year renewable contract - unlimited telephone and e-mail support from an assigned Customer Support Engineer; unlimited access to the web based knowledge base; access to all service releases (bug fixes) and engine upgrades (new functionality)

SoftWest currently uses Clarify as its company-wide problem reporting and tracking system and the Talk2Me Automated Voice Mail Data Collection System (AVM-DCS) for both screening and directing calls to Technical Support.

C.3.2 The requirements

3.0 Requirements

3.1 General

CSS should incorporate and/or interface to the Automated Voice Mail Data Collection System (AVM-DCS) information gathering, screening, and CallQ call director

CSS shall:

- Interface to and incorporate Clarify services for recording incidents reported electronically.
- Virus check any electronic inputs prior to any processing
- Screen voice and electronic input to determine the authorized level of service (Warranty, Star-Warranty, Silver, Gold, Bronze)
- Control the level of service provided via Clarify or via call direction
- Provide an immediate "dispute" escalation to Service Account Management if the customer asserts that he or she is entitled to a level of service that exceeds that recognized by CSS
- Provide management with the Clarify reports currently described in WWS Procedure 24-05 – with the addition of TBD reports and/or data incorporating all calls and electronic submissions.
- During screening and via Clarify store and protect customer-proprietary information, including any remote log in information, any operational data provided for trouble shooting, and other categories TBD
- Provide facilities for all personnel to report problems and authorized personnel to approve and perform data maintenance based on sales of new product, new contracts, contract renewals, and for correcting errors in recorded eligibility data
- Control SoftWest employee access to data – Global Summary (all accounts, all incidents, summary data TBD), Global Detail (all accounts, all incidents), at the Account level (all incidents), at the incident level. Control field-level permissions across all of the access categories (no access, read, write-add, write-revise)

3.2 Warranty

When it is determined that Warranty Service is authorized, CSS shall:

- Assign a priority and direct the telephone call to appropriate personnel. Note that WWS procedure 24-02 describes the various routing options and criteria. This includes follow-up calls where the caller is calling back at the request of a SoftWest Support Engineer.
- Assign a priority, set appropriate incident tracking and notification parameters, and record the data from the received electronic incident report in the Clarify data base. Note electronic report data may include attachments.

…

3.3 Star Warranty

CSS shall:

- Maintain a list of customers for whom Warranty Service is provided on an unlimited basis
- Automatically place all customers active at the time the CSS system and the associated service level contracts are implemented in the list
- Ignore the 12/12 parameters for customers in the list

When it is determined that Star Warranty service is authorized, CSS shall proceed as for Warranty Service.

3.3 Bronze

When it is determined that Bronze service is authorized, CSS shall:

…

3.4 Silver

When it is determined that Silver service is authorized, CSS shall:

…

3.5 Gold

When it is determined that Gold service is authorized, CSS shall:

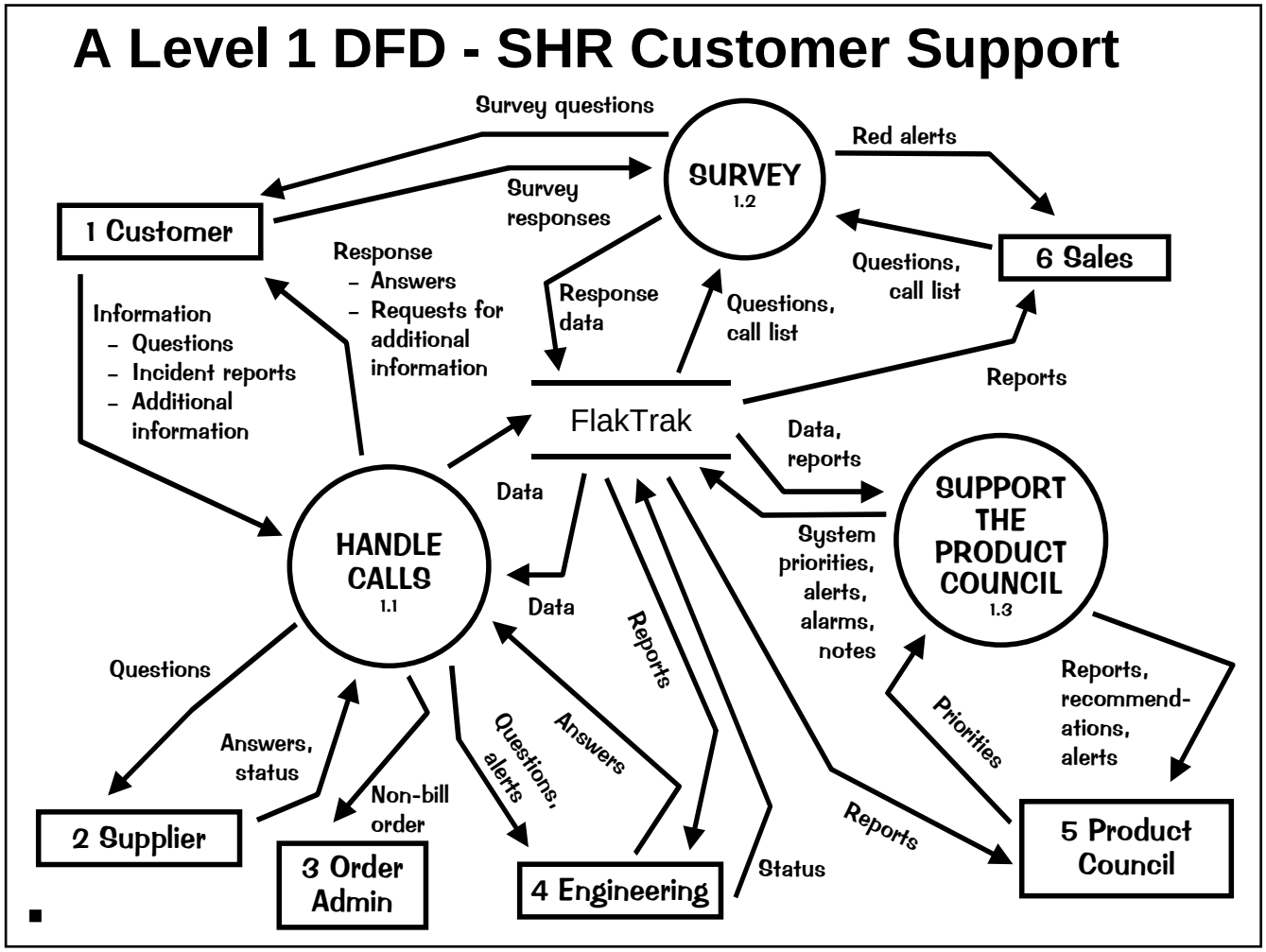
…

3.6 No service authorized

When it is determined that no service is authorized, CSS shall:

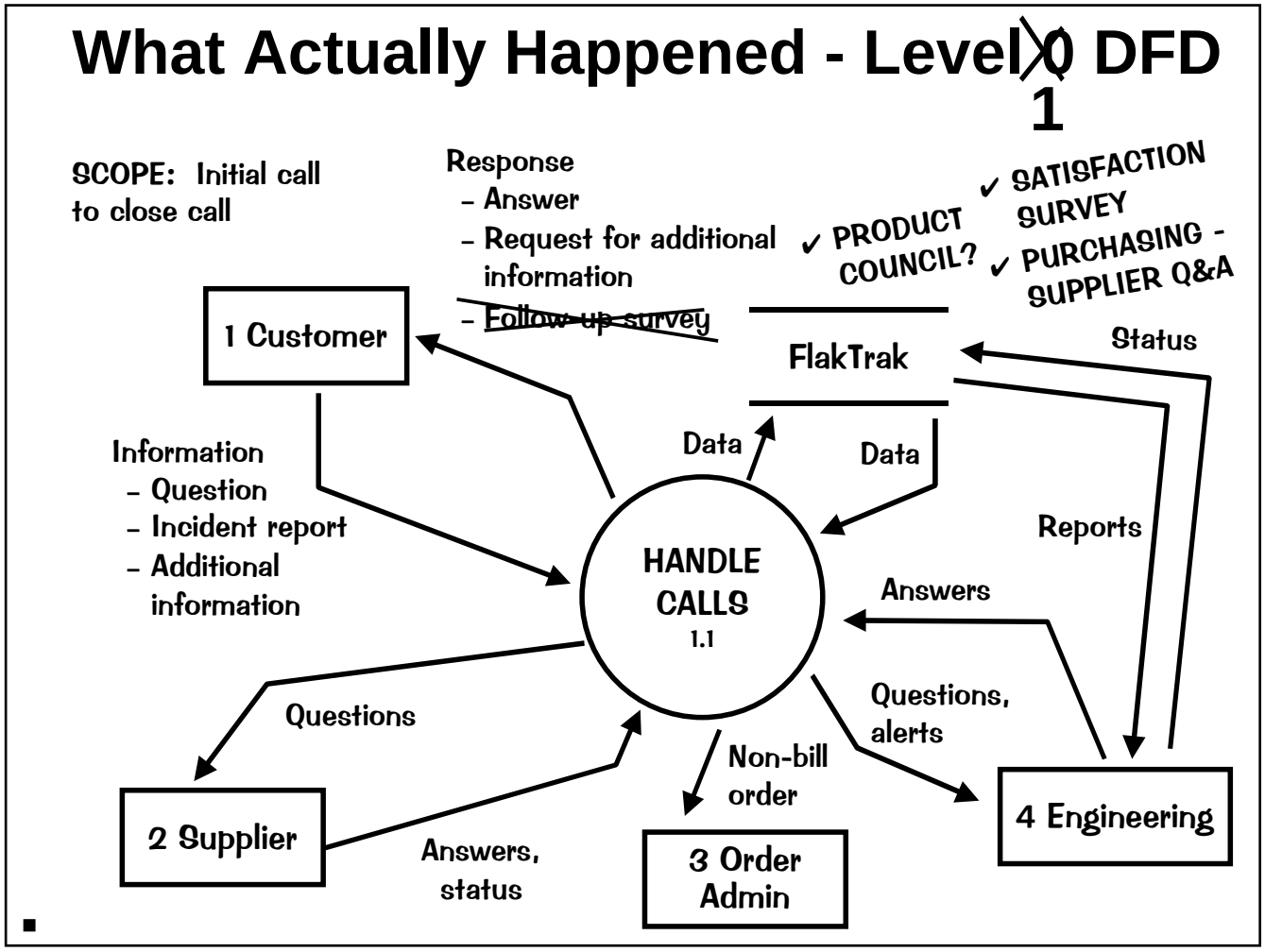
- Refer the report (telephone call or electronic report) to Service Contract Account Management (SCAM)
- Reprocess reports returned from the Manager of SCAM authorized for a particular level of service

…



The Question

Is this diagram consistent with the Level 0 DFD we created for SHR Customer Support? Is everything accounted for? Anything left out from the Level 0 DFD? Or added – that should also be in the Level 0 DFD?



The script

Person 1	[said with great authority and conviction] What we do is handle calls.
Person 2	Most important - in handling calls - we interact with the customer.
Person 1	We receive information - questions and incident reports and we ….
Person 2	… respond with answers to questions.
Person 3	Sometimes we have to ask for more information.
Facilitator	So, actually, I guess we better add "additional information" to what customers supply.
Person 2	Twice a year we also do the Customer Service Customer Satisfaction Survey.
Person 3	Wait just a minute. What are we talking about here? The survey is important, but what does it have to do with handling calls?
Facilitator	What we've been talking about is the process that goes from when a call comes in to when it's closed in the call handling system. I propose we take the survey off the table for now for now and and start a list so we'll be sure to come back to it.
All	(Nod heads in agreement.)

Person 1 Something else we do in handling calls is work with suppliers whose products we resell as part of our products. ☜ In a lot of cases, our customers don't even know it's not something we developed. They don't want to know. We ask ☜ questions ☜ and (sometimes) actually get ☜ answers or the current status ☜ of a problem they're working on for us. That's always a problem. Sometimes we'll start selling someone else's product and when the first call comes in and we call the supplier, it's like they're surprised.

Facilitator Let's add this to our list of follow-up items. ☜ I'll make sure the team working on how we procure products and services hears about the problem.

Person 3 We also spend a lot of time working with FlakTrak, our call handling and tracking system. ☜ We input data ☜ and access ☜ data ☜ to help check on status or research customer questions or problems.

Person 2 We also work with Engineering. ☜ We ask ☜ questions and give them alerts ☜ when we think they need to know about something immediately.

Person 3 Engineering answers our questions. ☜ They're usually pretty good about that, but sometimes – like around a new release – they're too busy.

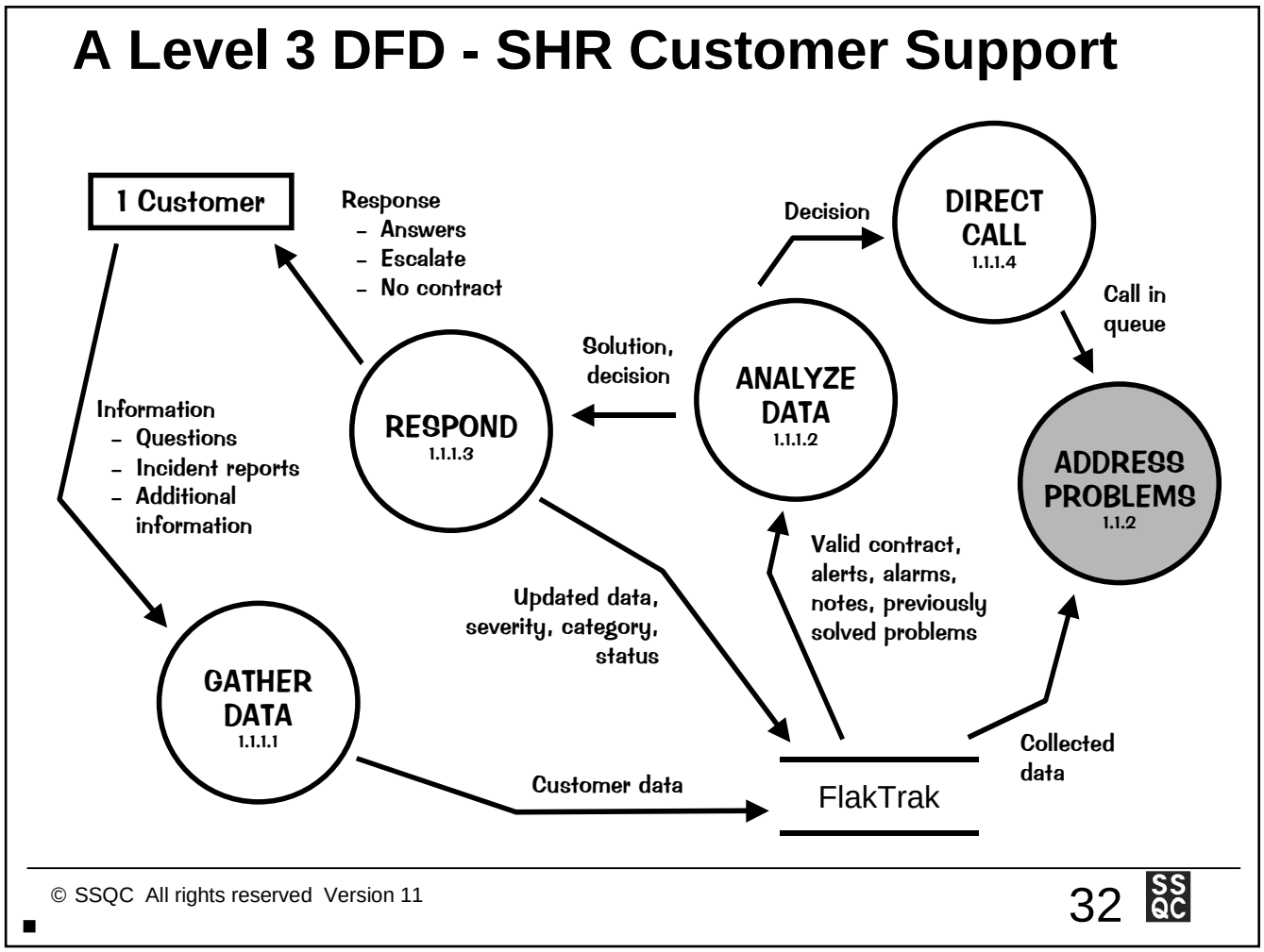
Person 1 By the way, Engineering has access to FlakTrak ☜ for pulling reports and they can also directly update ☜ the status of a problem or issue they're working.

Person 3 Sometimes, we have to work with Order Administration ☜. We'll initiate a non-bill order ☜ on the customer's behalf for an upgrade if that's the way we've determined we'll fix the problem.

Person 2 I don't think this qualifies as the top-level diagram anymore. We're going to have to go one step higher to add the survey. ☜

Person 4 While we're on the topic of other things we do, my biggest job is reporting to the Product Council and making sure that they add the features and make the changes we need to make the product supportable and keep our customers happy. We should probably add that to the follow-up list. ☜

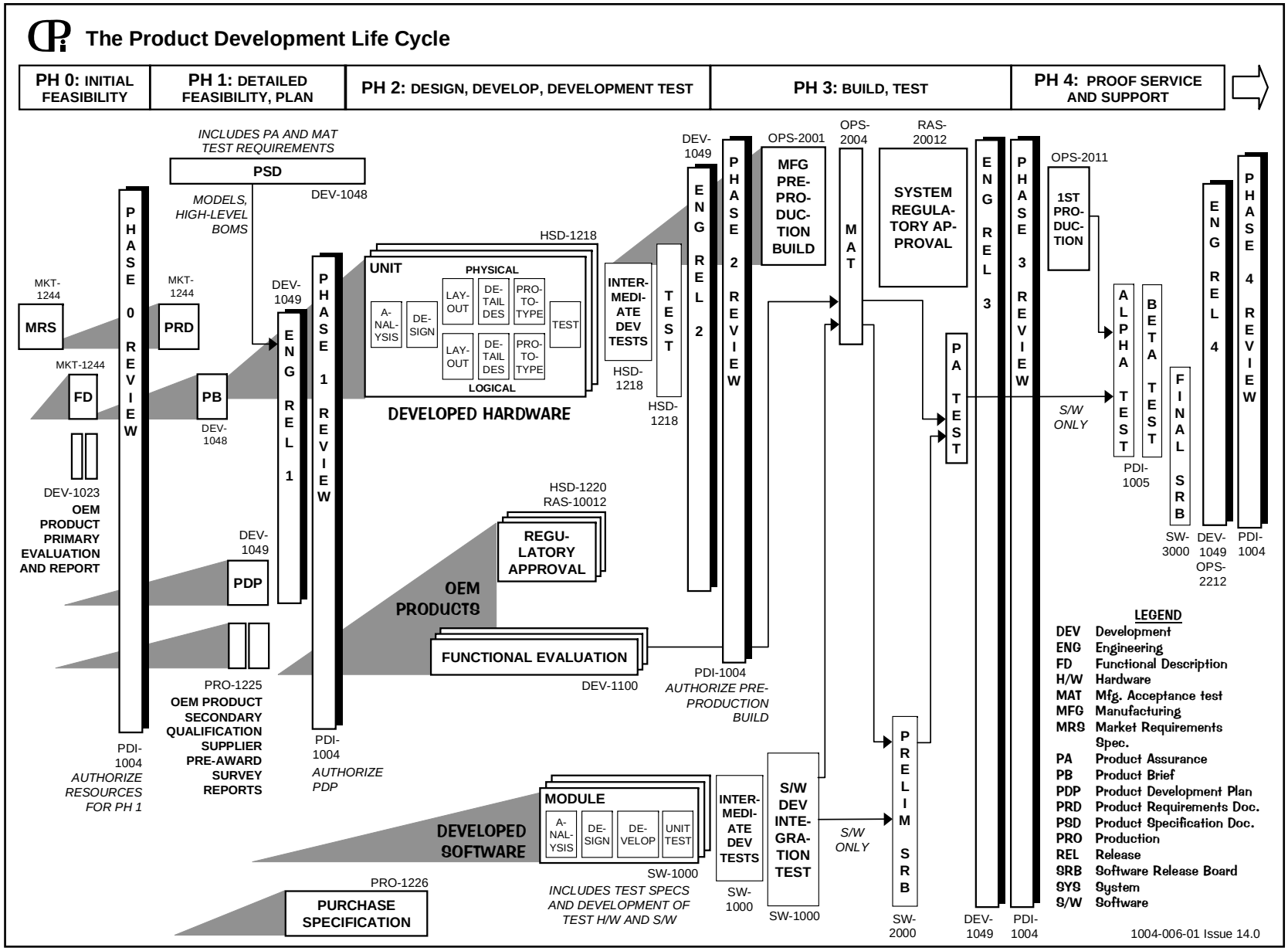
Facilitator ☜ Our time's just about up for this meeting – and I think we've got enough for me to go back and clean up my notes so we can pick up the discussion fresh next week. I'll have the diagrams to you by the day after tomorrow.



Exercise

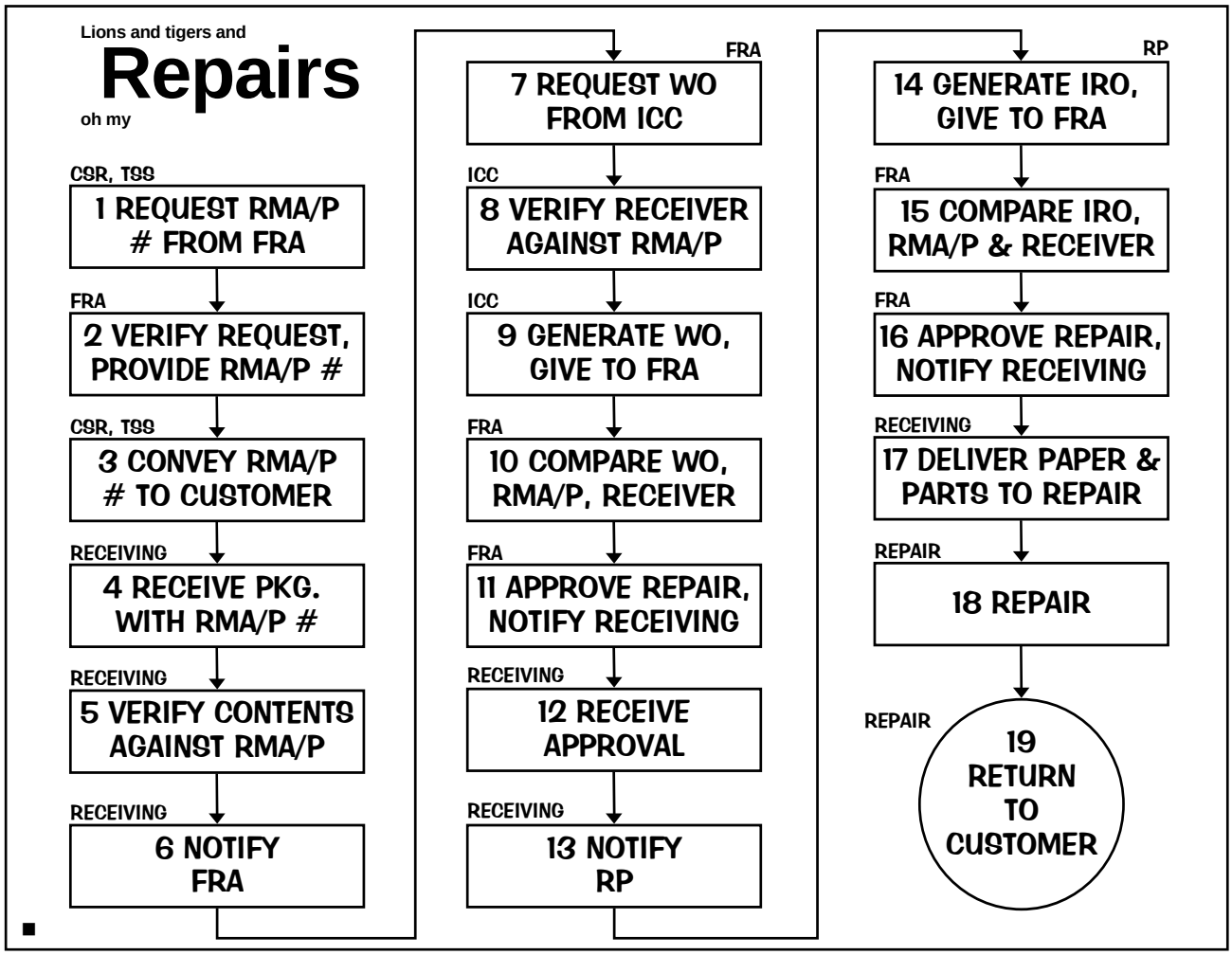
Based on the information you have on SHR, create a top-level DFD for the Order Administration function.

Choose one of the bubbles and decompose it.



Phone: +1-408-985-4476
FAX: +1-408-248-7772
info@ssqc.com http://www.ssqc.com

NOTES



CSR	**C**ustomer **S**upport **R**epresentative
FRA	**F**ield **R**eturn **A**dministrator
ICC	**I**nventory **C**ontrol **C**oordinator
IRO	**I**nternal **R**epair Order (adjusts inventory to account for item returned for repair)
RMA/P	**R**eturn **M**aterial **A**uthorization for Re**p**air (identifies the shipment as an authorized return)
RP	**R**epair **P**lanner
TSS	**T**echnical **S**upport **S**pecialist
WO	**W**ork **O**rder (puts the repair on the repair center schedule)

The Unified Modeling Language (UML)

The UML is a language for specifying, visualizing, constructing, and documenting the artifacts of software … [and] business … systems. It combines the work of Booch, Rumbaugh, Jacobson, and it is supported by the UML PARTNERS CONSORTIUM, made up of:

HP	IBM	ICON Computing
IntelliCorp	i-Logix	MCI Systemhouse
Microsoft	ObjecTime	Oracle
Platinum Technology	Ptech	Rational Software
Reich Technologies	Softeam	Sterling Software
Taskon	Unisys	

Rational Software is the Consortium administrator.

Background on the UML

During 1996, it became clear that several organizations saw UML as strategic to their business. A Request for Proposal (RFP) issued by the Object Management Group (OMG) provided the catalyst for these organizations to join forces around producing a joint RFP response. Rational established the UML Partners consortium with several organizations willing to dedicate resources to work toward a strong UML 1.0 definition. Those contributing most to the UML 1.0 definition included: Digital Equipment Corp., HP, i-Logix, IntelliCorp, IBM, ICON Computing, MCI Systemhouse, Microsoft, Oracle, Rational Software, TI, and Unisys. This collaboration produced UML 1.0, a modeling language that was well defined, expressive, powerful, and generally applicable. This was submitted to the OMG in January 1997 as an initial RFP response.

In January 1997 IBM, ObjecTime, Platinum Technology, Ptech, Taskon, Reich Technologies and Softeam also submitted separate RFP responses to the OMG. These companies joined the UML partners to contribute their ideas, and together the partners produced the revised UML 1.1 response. The focus of the UML 1.1 release was to improve the clarity of the UML 1.0 semantics and to incorporate contributions from the new partners. It was submitted to the OMG for their consideration and adopted in the fall of 1997.

OMG Background Information

The Object Management Group (OMG) was founded in April 1989 by eleven companies, including 3Com Corporation, American Airlines, Canon, Inc., Data General, Hewlett-Packard, Philips Telecommunications N.V., Sun Microsystems and Unisys Corporation. In October 1989, the OMG began independent operations as a not-for-profit corporation. Through the OMG's commitment to developing technically excellent, commercially viable and vendor independent specifications for the software industry, the consortium now includes over 800 members. The OMG is moving forward in establishing CORBA as the "Middleware that's Everywhere" through its worldwide standard specifications: CORBA/IIOP, Object Services, Internet Facilities and Domain Interface specifications.

Location and Sponsorships

The OMG is headquartered in Framingham, MA, USA and has international marketing offices in Australia, Bahrain, Brazil, Germany, India, Italy, Japan and the UK, along with a government representative in Washington, D.C. Additionally, the OMG is a sponsor of the COMDEX Enterprise series of Trade Shows and Conferences.

Mission

The OMG was formed to create a component-based software marketplace by hastening the introduction of standardized object software. The organization's charter includes the establishment of industry guidelines and detailed object management specifications to provide a common framework for

application development. Conformance to these specifications will make it possible to develop a heterogeneous computing environment across all major hardware platforms and operating systems.

Implementations of OMG specifications can be found on many operating systems across the world today. OMG's series of specifications detail the necessary standard interfaces for Distributed Object Computing. Its widely popular Internet protocol IIOP (Internet Inter-ORB Protocol) is being used as the infrastructure for technology companies like Netscape, Oracle, Sun, IBM and hundreds of others. These specifications are used worldwide to develop and deploy distributed applications for vertical markets, including Manufacturing, Finance, Telecoms, Electronic Commerce, Real-time systems and Health Care.

http://www.omg.org/omg

Tech Help desks wage internal war

Help desks duke it out

By Eve Tahmincioglu
New York Times

Debbie Bird, a help desk employee at the University of Cincinnati was prepared for rudeness from students and faculty members frustrated by frozen computer screens or software errors. But she never dreamed she would receive the same treatment from the computer wizards *she* had to turn to for help in solving her toughest problems.

"One woman downright yelled at me," Bird said about a software engineer who she said terrorized many of her call center colleagues. "She'd say, `Why are you calling? You should have handled this!' She was just pain nasty."

Bird is a Level 1 support technician, one of nearly 500,000 grunts in the United States who take your calls when your computer goes on the blink or stop by your desk to help you load new software. When their expertise fails them; they have to turn to Level 2 specialists, who make at least twice as much and sometimes have an attitude to match.

The disdain that Bird encounters is commonplace at help desks throughout the country, insiders say, with Level 1 personnel often disparaged as the hamburger flippers of the computer age.

Many managers are aware the friction and are trying to do something about it.

“The industry is struggling not to ghettoize support," said Jacque Rowden, operations manager for the Pittsburgh law firm of Buchanan Ingersoll. The people on the bottom of the hierarchy, she added, deserve "credibility and respect.”

Companies that do not take the clashes between the two groups seriously could end up paying a steep price, said Calvin Sun, founder of Technology Horizons, a consulting firm in Paoli, Pa., that trains technical-support workers. Level 1 technicians might be tempted to take out their frustrations on customers, he said, and their already sagging morale could sink even further.

These front-line workers, who make about $8 to $12 an hour, typically have little training and work long days. In addition to dealing with irate customers, many are under the gun by supervisors who monitor their calls and keep track of how much time they take to resolve problems. As a result, according to Steve Cain, benchmarking director for the help desk practice at the Gartner Group in Stamford, Conn., their turnover rate is close to 70 percent a year, one of the highest for any job category.

What really galls some of them are signs of incompetence from their supposed masters. Gianmarco Rossi, a Level 1 employee for a contractor in Orlando, Fla., recalled being unable to solve a customer's cable-modem problem, so he tried to forward the call. But the Level 2 technician refused to take it and instead proposed an endless stream of solutions, none of which worked. "I suppose the higher up you go, the lazier you are," Rossi said.

Some people worry that such mutual sniping can distract technicians of both Categories. Bill Artner, vice president for content development at Tech Republic of Louisville, Ky., which runs TechRepublic.com, says neither side should be made out as the bad guys. "Finding out what is truly wrong and not what the customer thinks is wrong is the tough part," Artner said. "Where Level 2 people get frustrated is when Level 1 techs don't accurately diagnose the problem. It's not an exact science and things can be misinterpreted. Sometimes the solution a Level 1 tech proposes can ultimately complicate the work for the Level 2."

Brian Bittner, a Level 2 technician at Dell Computer in Austin says he gets along well with Level 1 employees. He acknowledges that it gets under his skin, however, when they ask him basic technical questions or come back with a problem he has already gone over with them.

It probably helps that many people in Level 2 and even their bosses have worked in the trenches and know how grueling the experience can be. Everett Michaud, director of help desk services for Analysts International, an information technology concern in Minneapolis, says he just could not get any respect when he did time on the help desk at a Detroit automotive company in the early 1990s.

Every time Michaud showed initiative by asking questions that went beyond his immediate troubleshooting duties, he says, some higher-up would slap him down.

“You automatically become jealous and spiteful because you're put in a position where you can't even advance," he said.

With the bad old days in mind, he has taken a more democratic approach. He has gotten rid of the Level 1 and Level 2 categories, and he offers financial rewards to engineers and system specialists who go out of their way to be courteous to help desk workers. As a result, his help desk has a turnover rate of less than 5 percent, he said.

Michaud may have used persuasion to create a friendlier workplace, but sometimes a stick works best. Take the Level 2 technician who gave Bird and her colleagues at the University of Cincinnati a hard time. Clarence Smith, manager of operations at the school's help desk, finally confronted her and threatened to report her conduct to her supervisor. Since then, he says, she has mended her ways.

"They don't have to love us," he said, "but they've got to try to get along with us."

Phone: +1-408-985-4476
FAX: +1-408-248-7772
info@ssqc.com http://www.ssqc.com

FROM: H. Strasser, SAV Marketing Manager CP 03.004.01

TO: Engineering

VIA: B. Arnold, VP Marketing

SUBJECT: Modifications to SAV

REF: MRT-COM03 Presentation, 19 Feb, at Bethesda

ENCL: None

Background

There is a significant need to extend the application of technology developed and successfully deployed for the ship-board automated Spaces and Voids (SAV) Monitoring system. The released, deployed systems have sensors in the 155X SmartSensor family for intrusion (infrared – 1551, sound – 1552), fire (1553), temperature (1554), and humidity (1555). These sensors are hard wired to a SmartSensor LED panel, which offers a fixed, limited functionality. The panel firmware (not field modifiable) controls the reading or change in reading at which the LED corresponding to a sensor is illuminated and at which an audible (horn) and/or visible (light) alarm is activated. The points at which the light/alarm are triggered are set for each sensor (based on the pre-installation survey) when the firmware for the particular installation is burned. Each sensor is checked periodically by attaching a SmartSensorTestMeter to the test points on the sensor.

The PDI K9 ruggedized personal computer and battery power-supply (PDI UPS) are compatible with the S-BUS architecture. Any 155X sensor can be connected directly to an S-BUS I/O card. All three versions (ISA, PCI, or AGP) of the S-BUS I/O card that can be installed in the K9 can handle 12 sensors. The K9 has 3 open slots. The K9 can also be configured with up to three chassis extenders that will also accept the S-BUS I/O cards. Each chassis extender takes up one open slot in the K9 and adds slots for up to 4 S-BUS cards. The Coast Guard and SDCC (StarFleet Discount Caribbean Cruises) already deploy 1553 sensors with the K9 as beta versions of the Model 64 FDS (Fire Detection System) on ships in their fleets. The Model 64 issues alerts at the PC (operator console), which are then acted upon by whoever is responsible for monitoring the K9. Marketing had a contractor throw together a VisualBasic application to process the sensor inputs and issue the alerts from the K9.

In addition, YourAttic, Inc., which rents storage space on a monthly basis at very competitive prices, has standard SAV systems currently installed at five of its facilities with the panel in the Manager's Office.

The Market Requirements

What the market requires (see the data from the MRT-COM03 presentation) is a soft, user-configurable version of the hard wired SAV system for both ship-board and shore-based applications. It should probably

be driven from a PC with on-site user configurability. The market requires a PC-based system to have all the features of the hard-wired system (alarms, all different kinds of sensors, defined thresholds at which alarms are triggered – both for absolute values from the sensors and for rate of change over time for appropriate kinds of sensors, etc.).

The system should be able to fire off alerts about sensor status from the PC – and to be able to interface to AutoCall4Help to the AFSS (Automated Fire Suppression System).

Based on demographics and the competition, the largest warehouse facility in the target market will have 400,000 square feet of floor space. Based on IEEE-SPEC 1502/4 (Sensor Positioning for Optimum Detection), the system will need to support up to 196 sensors. Since the system will be going into warehouses and large open spaces that can have multiple, changeable layouts with varying contents, users require the ability to define up to 6 zones, where a zone is a collection of up to 54 sensors. This is more than will ever be needed for an install on a ship.

Due to the nature of the environments we anticipate serving with this system, customers (especially current users) still want the hard-wired panel as an off-line back-up in case the K9 fails. The system will need to provide 24 hours of off-line operation from the hard-wired panel. The K9 should operate for an hour. In at least the first phase, off-line operation can revert to the functionality currently available in the hard-wired, panel system (e.g., no zones). For future releases, we can get hardware involved and evaluate the feasibility of adding some sort of zone logic to the panel.

In addition to the configurability described above, customers require that, from the console, an authorized operator be able to poll the sensors, check status, and test operational status of the sensors, which is something the hard-wired panel can't do. Could such an operational status check be programmed into the panel firmware for a future release?

To keep budgets and time frames reasonable, the solution should require no hardware development.

	Source	Requirement

	Requirement							

Requirements Management Tools References

Product	Contact
Alta SPW	Cadence Design Systems 555 N. Mathilda Ave. Sunnyvale, CA 94086 http://www.cadence.com
Caliber-RM	Technology Builders, Inc. 400 Interstate North Parkway, Suite 1090 Atlanta, GA 30339 Phone: 800-937-0047 Fax: 770-937-7901 http://www.tbi.com/products/caliber.html
CORE	Vitech Corporation 2070 Chain Bridge Rd Suite 320 Vienna, VA 22182-2536 http://www.vtcorp.com
Cradle	Structured Software Systems P.O. Box 310 Olney MD 20830-0310 Structured Software Systems Craven House Michaelson Road Barrow-in-Furness Cumbria LA14 2RJ UK http://www.threesl.com
DOORS	Quality Systems & Software (QSS) North American Headquarters 200 Valley Road Suite 306 Mt. Arlington New Jersey 07856 Tel: +1 973 770 6400 Fax:+1 973 770 6401 http://www.qssinc.com/home.cfm
Extend	Imagine That, Inc. 6830 Via Del Oro, Suite 230 San Jose, CA 95119 http://www.imaginethatinc.com
Foresight	Nu Thena Systems, Inc 11824 Jollyville Road, Suite 101 Austin TX 78759 http://www.nuthena.com/
MetricCenter	2300 Fall Hill Avenue, Suite 100 Fredericksburg, VA 22401 http://www.distributive.com
RDD.COM	Ascent Logic Corporation 180 Rose Orchard Way Suite 200 San Jose, CA 94134 http://www.alc.com

Product	Contact
RDT	GEC Macron Systems PTA Limited 40-52 Talavera Rd, North Ryde, NSW 2113 AUSTRALIA http://www.ausnet.net.au/gecm/
RequisitePro	Rational Software Corp. 18880 Homestead Rd Cupertino CA 95014 http://www.rational.com
RTM	Integrated Chipware Inc. 1861 Wiehle Avenue, Suite 300 Reston, VA 20190 http://www.chipware.com
SLATE REquire	TD Technologies 2425 N. Central Expressway, Suite 200 Richardson, TX 75080 http://www.tdtech.com
Statemate MAGNUM	I-Logix, Inc. Three Riverside Drive Andover MA 01810 http://www.ilogix.com
Tofs	Tofs AB Fridhem 2 S-76040 Veddoe Sweden http://www.toolforsystems.com
VitalLink	Compliance Automation, Inc. 2703 West Long Drive Unit C Littleton CO 80120 http://www.complianceautomation.com
XTie-RT	Teledyne Brown Engineering 300 Sparkman Dr. Cummings Research Park PO Box 070007 Huntsville, Alabama 35807-7007 http://www.tbe.com/products/xtie/xtiertprod.html

Mapping Tools References

ABC FlowCharter	Business drawing and diagramming tool, http://www.micrografx.com
allClear	From SPSS, flow charting tool with links to Clear Process, http://www.spss.com/software/allclear/
Argo/UML	A free UML modeling tool, http://www.argouml.com
Bpwin	From LogicWorks, now part of Platinum Technologies, now part of Computer Associates, modeling tool used to analyze, document, and improve business processes, http://www.platinum.com
CorelFlow	From Corel, a diagramming tool, http://www.corel.com
GDPro	From Embarcadero Technology, visual UML modeling tool, includes design and code generation (Java, C++, IDL), priced as such, www.embarcadero.com
Inspiration	From Inspiration Software, Inc., for concept maps, web maps, idea organization, http://www.inspiration.com (free 30 day trial available)
MQSeries Workflow	Formerly FlowMark, from IBM, www.software.ibm.com/ad/flowmark
Optima	Integrated tool for creating presentation-quality process maps, modeling process behavior, doing simulation, and performing "what if?" analysis, http://www.micrografx.com/enterprise/optima/
Process98	From Scitor, the next generation of ProcessCharter, tool to design, simulate, and improve business and manufacturing processes, http://www.scitor.com
ProcessWise WorkBench	From International Computers Ltd. (ICL), now part of Fujitsu, modeling, simulation, and analysis tool, http://www.teamware-us.com/products/others.htm
TeamFlow	Process diagramming tool from CFM Inc., www.teamflow.com
Visio Standard	From Visio Corporation, a diagramming tool, www.visio.com

NOTES

Requirements Engineering References

Ref	Internet	Title and Description
ACA1	http://www.qssinc.com	Acaba, Ralph H., **Lessons Learned in the Selection of a Company Standard Requirements Management Tool** (see QSS1)
AMB1	http://www.sdmagazine.com/articles/2000/0006/0006j.htm	Ambler, Scott, *Object-Oriented Business Rules*, **Software Development**, June 2000. Key Concepts Business Rules are a key element for defining requirements and designing systems.
BCS1	http://research.ivv.nasa.gov/~steve/resg/	Requirements Engineering Quarterly, from the Requirements Engineering Specialist Group of the British Computer Society (current as of 1996)
CAI1	http://www.complianceautomation.com	The home page of Compliance Automation, Inc. includes references to several papers. Of particular interest are FEL1, HOO1, HOO2, and HOO3.
CRE1		Creel, Chris, *Requirements by Pattern*, **Software Development**, Vol. 7, No. 12, December 1999, page 44
DAV1	http://www.rational.com/sitewide/media/696wp.pdf	Davis, Alan M.; Leffingwell, Dean A., **Using Requirements Management to Speed Delivery of Higher Quality Applications**, 1996, Rational Software Corporation
DAV2	http://mozart.uccs.edu/adavis/reqbib.html	Requirements management bibliography
DIO1	http://www.stsc.hill.af.mil/CrossTalk/1994/feb/xt94d02c.asp	Dion, Raymond, *Process Improvement and the Corporate Balance Sheet*, **IEEE Software**, volume 10, number 3, pages 28-35, July 1993
DOE1	http://cio.doe.gov/smp	US Department of Energy (DOE), Software Management Program home page. Of particular interest is the **Software Engineering Methodology** (SEM), *Chapter 4, Requirements Definition Stage*
DOL1	http://www.stsc.hill.af.mil/crosstalk/1994/jan/xt94d01g.asp	Dolan, Kevin, *Prototypes: Tools That Can Be Used and Misused*, **CrossTalk**, January, 1994
EIA1	http://www.geia.org/eoc/G47/731dwnld.htm	Electronic Industries Alliance (EIA), **EIA/IS-731.1, Systems Engineering Capability Model (SECM)**, and **EIA/IS-731.2, SECM Appraisal Method**
ELL1	http://www.cs.ucl.ac.uk/staff/W.Emmerich/publications/CACM/nats.html	Ellmer, Ernst; Emmerich, Wolfgang; Finkelstein, Anthony; Galal, Galal, **Improving Requirements Management**, 1998
EMM1	http://www.cs.ucl.ac.uk/staff/W.Emmerich/publications/CACM/tools.html	Emmerich, Wolfgang; Finkelstein, Anthony; Stevens, Richard, **The Future of Requirements Management Tools**, 1998
EVA1	See http://www.stsc.hill.af.mil/crosstalk – if available	Evans, Michael W. *SPMN Director Identifies 16 Critical Software Practices [for Performance-Based Management]*, **CrossTalk**, March 2001, page 27
FEL1	http://www.complianceautomation.com	Fellow, Larry; Hooks, Ivy, **A Case for Priority Classifying Requirements** (see CAI1)

Ref	Internet	Title and Description
[FIN1]	http://www-dept.cs.ucl.ac.uk/staff/A.Finkelstein/publb.html	Gotel, O. & Finkelstein, A.; "An Analysis of the Requirements Traceability Problem" in **Proceedings of the 1st International Conference on Requirements Engineering** 1994, (IEEE CS Press) 1994, 94-101. (rtprob.ps.gz) There are a number of additional, relevant articles at this site. Nuseibeh, B., Kramer, J. & Finkelstein, A. "A Framework for Expressing the Relationships Between Multiple Views in Requirements Specification", **IEEE Transactions on Software Engineering**, 20, 10 (1994), 760-773. (tse94.icse.ps.gz) Finkelstein, A. "Requirements Engineering: a review and research agenda" in **Proceedings of the 1st Asian & Pacific Software Engineering Conference**, (IEEE CS Press) 1994, 10-19. (rereview.ps.gz) Gotel, O. & Finkelstein, A.; "Contribution Structures" in **Proceedings of the 2nd International Symposium on Requirements Engineering RE95**, (IEEE CS Press) 1995, 100-107 (contrib.ps.gz) Gotel, O. & Finkelstein, A. "Extended Requirements Traceability: results of an industrial case study" in **Proceedings of the 3rd International Symposium on Requirements Engineering RE95**, (IEEE CS Press), 1997, 169-178. (casetrace.ps.gz)
GEN1	http://ricis.cl.uh.edu/virt-lib/requirements.html	An RM bibliography
GEN2	http://www.ida.liu.se/labs/aslab/people/joaka/re_bib.html	A requirements engineering bibliography
HAM1	http://www.stsc.hill.af.mil/crosstalk/1998/dec/hammer.asp	Hammer, Theodore; Huffman, Leonore L., Rosenberg, Linda H., *Doing Requirements Right the First Time*, **CrossTalk**, December 1998
HOF1	http://www.ifi.unizh.ch/techreports/	IFI – Institut fur Informatik der Universitat Zurich – technical reports library. Of particular interest is Technical Report Nr. 93-5, Hofmann, Hubert, **Requirements Engineering, A Survey of Methods and Tools**, March 1993. To find the report, go to the URL listed above. Below the heading, “Index of Electronically Available Technical Reports”, select “1993 Technical Reports”. From the list of 1993 reports, select ifi-93.05 for a copy of the report.
HOO1	http://www.complianceautomation.com	Hooks, Ivy, **Writing Good Requirements** (writingreqs.html), from the 4th INCOSE Symposium (see CAI1)
HOO2	http://www.complianceautomation.com	Hooks, Ivy, **Why Johnny Can’t Write Requirements** (whyjohnny.html), from the 1990 AIAA Conference (see CAI1)

Ref	Internet	Title and Description
HOO3	http://www.complianceautomation.com	Hooks, Ivy, **Managing Requirements** (see CAI1)
IEE1	http://www.ieee.org (to purchase)	IEEE Std 830-1998, **IEEE Recommended Practice for Software Requirements Specifications**, Institute of Electrical and Electronics Engineers, Inc., New York, 1998, ISBN 0-7381-0332-2 [Note this is a recommended practice. It is currently under considerations for revision/reissue as a standard.]
IEE2	http://www.ieee.org (to purchase)	IEEE Std. 610.12-1990, **IEEE Standard Glossary of Software Engineering Terminology**, Institute of Electrical and Electronics Engineers, Inc., New York, 1990 (corrected 1991), ISBN 1-55937-067-X
IEE3	http://www.ieee.org (to purchase)	IEEE Std. 1233-1998, **IEEE Guide for Developing System Requirements Specifications**, Institute of Electrical and Electronics Engineers, Inc., New York, 1998
IEE4	http://www.ieee.org (to purchase)	IEEE/EIA 12207.0-1996 **Software Life Cycle Processes**, Institute of Electrical and Electronics Engineers, Inc., New York, 1998, ISBN 1-55937-977-4 [12207.0 (part 0) is the same as ISO/IEC 12207. IEEE also offers parts 1 (12207.1) and 2 (12207.2)with additional guidance.]
INC1	http://www.incose.org/tools/tooltax.html	**Tools Taxonomy: Requirements Management Tools**, International Council on Systems Engineering (INCOSE), 1999
INC2	http://www.incose.org/stc/news731.htm	**EIA/IS 731 Systems Engineering Capability Model**
ISO1		ISO/IEC 15288 **System Engineering - System Life Cycle Processes** (final committee draft; not yet publicly available)
JAC1		Jackson, Michael, **Problem Frames and Methods: Structuring and Analyzing Software Development Problems**, 1st ed, Addison Wesley Longman, Inc., September 2000, ISBN: 020159627X
KAR1	http://www.incose.org/workgrps/rwg/goodreqs.html	Kar, Pradip; Bailey, Michelle, **Characteristics of Good Requirements**, 1996 Symposium, International Council on Systems Engineering (INCOSE), 1996
LAN1	http://www.comp.lancs.ac.uk/computing/research/cseg/index.html	The homepage of the University of Lancaster, Cooperative Systems Engineering Group (CSEG) has numerous articles and free tools.
LEF1	http://www.rational.com/products/reqpro/prodinfo/whitepapers/reqpro_index.jtmpl	Leffingwell, Dean A., **A Field Guide to Effective Requirements Management under SEI's Capability Maturity Model**, 1996, Rational Software Corporation (see RAT1)
MCC1	http://www.sdmagazine.com/articles/1996/0008/0008a/0008a.htm	McConnell, Steve, *Software Quality at Top Speed*, **Software Development**, August 1996, page 38 Key Concepts: "[While] Some project managers try to shorten ... schedules by reducing quality

Ref	Internet	Title and Description
		assurance practices such as design and code reviews ... [studies show that] projects that achieve the lowest defect rates also achieve the shortest schedules." (page 39) The figures are not included in the online version, but the verbal description of Figure 1 identifies the 95% defect removal level as optimum for reducing development time. (page 40) "Reworking defective requirements, design, and code typically consume 40% to 50% of the total cost of software development." (page 41) "Every hour you spend on defect prevention will reduce repair time from three to ten hours." (page 41) "Reworking a ... requirements problem once the software is in operation typically costs fifty to two hundred times what it would take to rework the problem in the requirements stage." (page 41) "... about 60% of all defects usually exist by design time." (page 41) See the section on "Additional Reading" in the side bar at the end of the article, on page 42.
NOV1	http://www.qssinc.com	Novorita, Robert J.; Grube, Gary, **Benefits of Structured Requirements Methods for Market-Based Enterprises** (see QSS1)
OBE1	http://www.rational.com/products/reqpro/prodinfo/whitepapers/reqpro_index.jtmpl	Oberg, Roger; Probasco, Leslee; Ericsson, Maria, **Applying Requirements Management with Use Cases**, Rational Software Corporation, 1998 (see RAT1)
QSS1	http://www.qssinc.com	After a free registration, you can download papers from the on-line library. When you read or download a paper, note that the papers are spread across multiple files (follow the "see next page" link at the end of the file). Of particular interest are: ACA1, NOV1, RAY1, STE1, STE2
QSS2		**QSSNewsByte**, an RM journal published electronically by QSS, Inc. (supplier of DOORS). To subscribe or obtain more information, contact: http://www.qssinc.com/lists/qssnewsbyte/subscribe.cfm
RAM1	http://www.stsc.hill.af.mil/crosstalk/1995/apr/lessons.asp	Ramesh, Bala; Stubbs, Curtis (Lt.); Powers, Timothy (LCDR); Edwards, Michael, *Lessons Learned from Implementing Requirements Traceability*, **CrossTalk**, April 1995
RAT1	http://www.rational.com/products/reqpro/prodinfo/whitepapers/reqpro_index.jtmpl	After a free registration you can download white papers on requirements management. Of particular interest are LEF1, OBE1.
RAY1	http://www.qssinc.com	Raymond, Paul, **A Comparison of Two Approaches to User Interfaces for Requirements Management and Traceability Tools** (see QSS1)

Ref	Internet	Title and Description
SEI1	http://www.sei.cmu.edu/publications/documents/95.reports/95.mm.003.html	Bate, Roger, et al., **A Systems Engineering Capability Maturity Model, Version 1.1, SE–CMM**, SECMM-95-01, CMU/SEI-95-MM-003, Carnegie Mellon University, Software Engineering Institute, November 1995. *PA 06* and *PA 02* are of particular interest.
SEI2	http://www.sei.cmu.edu/publications/documents/93.reports	Paulk, Mark C., et al., **Key Practices of the Capability Maturity Model , Version 1.1**, CMU/SEI-93-TR-025, Carnegie Mellon University, Software Engineering Institute, February 1993
SEI3	http://www.sei.cmu.edu/publications/documents/02.reports/02tr002.html	**Capability Maturity Model Integrated (CMMI) for Systems Engineering/ Software Engineering, Version 1.1**, CMU/SEI-2002-TR002
SOM1		Sommerville, Ian; Sawyer, Pete; Sommerville, Aan, **Requirements Engineering: A Good Practice Guide**, John Wiley & Sons; ISBN: 0471974447; 1997
STA1	http://www.standishgroup.com/chaos.html	The Standish Group's paper, **Chaos**, on failures of software projects, 1995.
STA2	http://www.standishgroup.com/voyages.html	The Standish Group's paper, **Unfinished Voyages**, 1996
STE1	http://www.qssinc.com	Stevens, Richard; Putlock, Gary, **Improving Requirements Management** (see QSS1)
STE2	http://www.qssinc.com	Stevens, Richard; Martin, James, **What is Requirements Management?** (see QSS1)
VAN1	http://www.stsc.hill.af.mil/crosstalk/1998/dec/cook.asp	Van Buren, Jim; Cook, David A., *Experiences in the Adoption of Requirements Engineering Technologies*, **CrossTalk**, December 1998
WAT1	http://www.rational.com/products/reqpro/prodinfo/media/softreq.pdf	Waters, John K., *Software Requirements Management, Five Tools up to the Task*, from **Component** Strategies, April 1999, (www.componentmag.com)
WEI1		Weinberg, Gerald M., **Quality Software Management, Volume 2, First Order Measurement**, Dorset House Publishing, NY, 1993
WIE1	http://www.sdmagazine.com/supplement/ppm/features/s997ppm1.shtml	Wiegers, Karl, *Automating Requirements management*, **Software Development Magazine**, Vol. 7 , No. 7, July 1999

NOTES

Mapping References

[BO1] Booch, Grady, Rumbaugh, James, Jacobsen, Ivar, **The Unified Modeling Language User Guide**, Addison-Wesley, Reading Massachusetts, 1999, ISBN 0-201-57168-4

[FO1] Fowler, Martin, Scott, Kendall, **UML Distilled, Applying the Standard Object Modeling Language,** Addison-Wesley, Reading Massachusetts, 1997 (10th printing, December 1998), ISBN 0-201-32563-2

[GA1] Gardner, Robert A., *Resolving The Process Paradox*, **Quality Progress**, March 2001

[IM1] Imai, Masaaki, **KAIZEN, The Key to Japan's Competitive Success**, McGraw-Hill Publishing Company, New York, 1986, ISBN 0-07-554332-X

[JA1] Jackson, Michael, **Problem Frames**, Addison-Wesley, ACM Press (Pearson Education), Harlow England, 2001, ISBN 0-210-59627-X

[JU1] J. M. Juran, **Juran on Planning for Quality**, The Free Press (Macmillan, Inc.), New York, 1988, ISBN 0-02-916681-0 [pages 18 - 22]

[NI1] (NIST), **Integration Definition for Function Modeling (IDEF0)**, National Institute of Standards and Technology; available from 2 on-line sources:
ftp://ftp.dtic.mil/pub/bpr-help desk/document/fips183.rtf
http://nemo.ncsl.nist.gov/idef/standsp /idef0.html

or order FIPSPUB 183 from:
National Institute of Standards and Technology
Department of Commerce
Springfield VA 22161

[OM1] (OMG), **UML Notation Guide**, **UML Semantics**, **UML Extension for Business Modeling**; these three volumes and several others are available from the Object Management Group on-line at
http://www.omg.org/techprocess/meetings/schedule/Technology_Adoptions.html#tbl_UML_Specification

Note that this address is entered on a single line.

[PR1] Roger S. Pressman, **Software Engineering, A Practitioner's Approach**, 3rd ed., Mc-Graw Hill Inc., 1992, New York, ISBN 0-07-050814-3

[YO1] Edward Yourdon, **Modern Structured Analysis**, Yourdon Press/P T R Prentice Hall, 1989

NOTES

204th printing x6.K_KO.mmb.qty (50K)

Software Systems Quality Consulting
2269 Sunny Vista Drive, San Jose CA USA
+1-408-985-4476 http://www.ssqc.com

Software ◆ Systems ◆ Quality ◆ Consulting
2269 Sunny Vista Drive ◆ San Jose CA USA ◆ Tel +1-408-985-4476
info@ssqc.com ◆ www.ssqc.com