Value Proposition for Project Managers from Configuration Management

CMMI 2008
Denver, Colorado
United States
November 2008

Tim Kasse
Kasse Initiatives LLC
+1 – 972 - 987 - 7706 USA
+45 72 19 42 18 Europe

Welcome

ようこそ

WelKom

Huan Yín

Bienvenido

Bienvenue

Wilkommen

Κ́Α Λ Ο Σ Ο Ρ Ι Σ Α Τ Ε

Bienvenuto

Välkommen

Tervetuloa

Witamy

ברוכים הבאים

Do You Like Configuration Management?

- ***I Hate Configuration Management!***
- ***But…………………………***
- ***15 years of rework and financial loss data later……………….***

Do You Like Configuration Management? - 2

- ***Let's talk about what CM is supposed to do for you as a Project Manager and more than that…..***
- ***What should you demand it do for your project***

What Is Configuration Management?

Purpose of Configuration Management

- The purpose of Configuration Management is to establish and maintain the integrity of work products using configuration identification, configuration control, configuration status accounting, and configuration audits.

from Capability Maturity Model® Integration (CMMI® - DEV V 1.2")

Configuration Management

- Identifying and defining the configuration items in a product
- Controlling the baselines and proposed changes of the configuration items and other related work products in those baselines throughout the product life cycle
- Recording and reporting the status of configuration items and change requests
- Verifying the completeness and correctness of configuration items through configuration auditing
- Preparing a product or product component for release

The Need for CM (S/W Example)

- The most frustrating software problems are often caused by poor configuration management:
  - The latest version of source code cannot be found
  - A difficult bug that was fixed at great expense suddenly reappears
  - A developed and tested feature is mysteriously missing
  - A fully tested program suddenly does not work
  - The wrong version of the code was tested

The Need for CM (S/W Example) - 2

- There is no traceability between the software requirements, documentation and code
- Programmers are working on the wrong version of the code
- The wrong version of the configuration items is being baselined
- No one knows which modules comprised the software system delivered to the customer
- Rework is performed!

Key CM Activities

Configuration Identification

- Identifying the structure of the system
- Identifying all related life-cycle work products, designated internal work products, acquired products, tools, and other items that are used in creating and describing these work products
- Providing a unique identifier for each of those work products
- Supporting bi-directional traceability from the customer requirements to the product and product component requirements to Systems Test and back
- Supporting traceability to all related plans such as the Project Plan, the Project Quality Plan, the Risk Management Plan, etc.

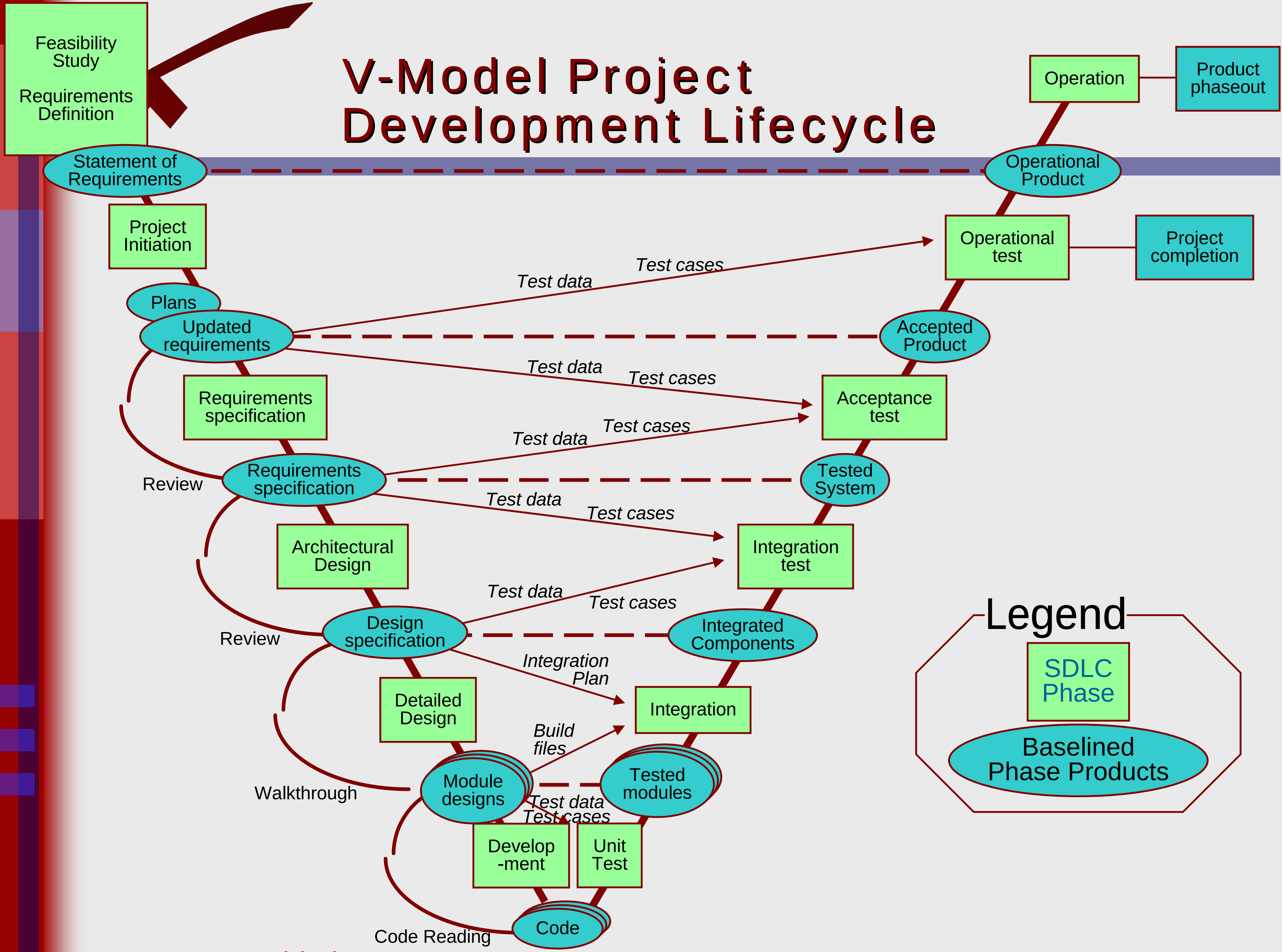
Examples of Configuration Items

Examples of work products that may be identified to be placed under configuration control include:

- Requirements Specification
- Architecture Specification
- Interface Specifications
- Design Specifications
- Product Specifications
- Drawings
- Code Modules
- Project Plan
- Quality Plans
- Configuration Management Plan
- Risk Management Plan
- Test Plans
- Test Procedures
- Development Procedures
- Standards
- Product Technical Publications
- Compilers
- Operating Systems
- Linkers/Loaders
- Procedure Languages
- Shell Scripts
- Other related support tools
- Third Party Tools
- Data Dictionaries
- System Build Files
- Product Data Files
- Logical Data Structures
- User Interface files, data
- Installation / Configuration files

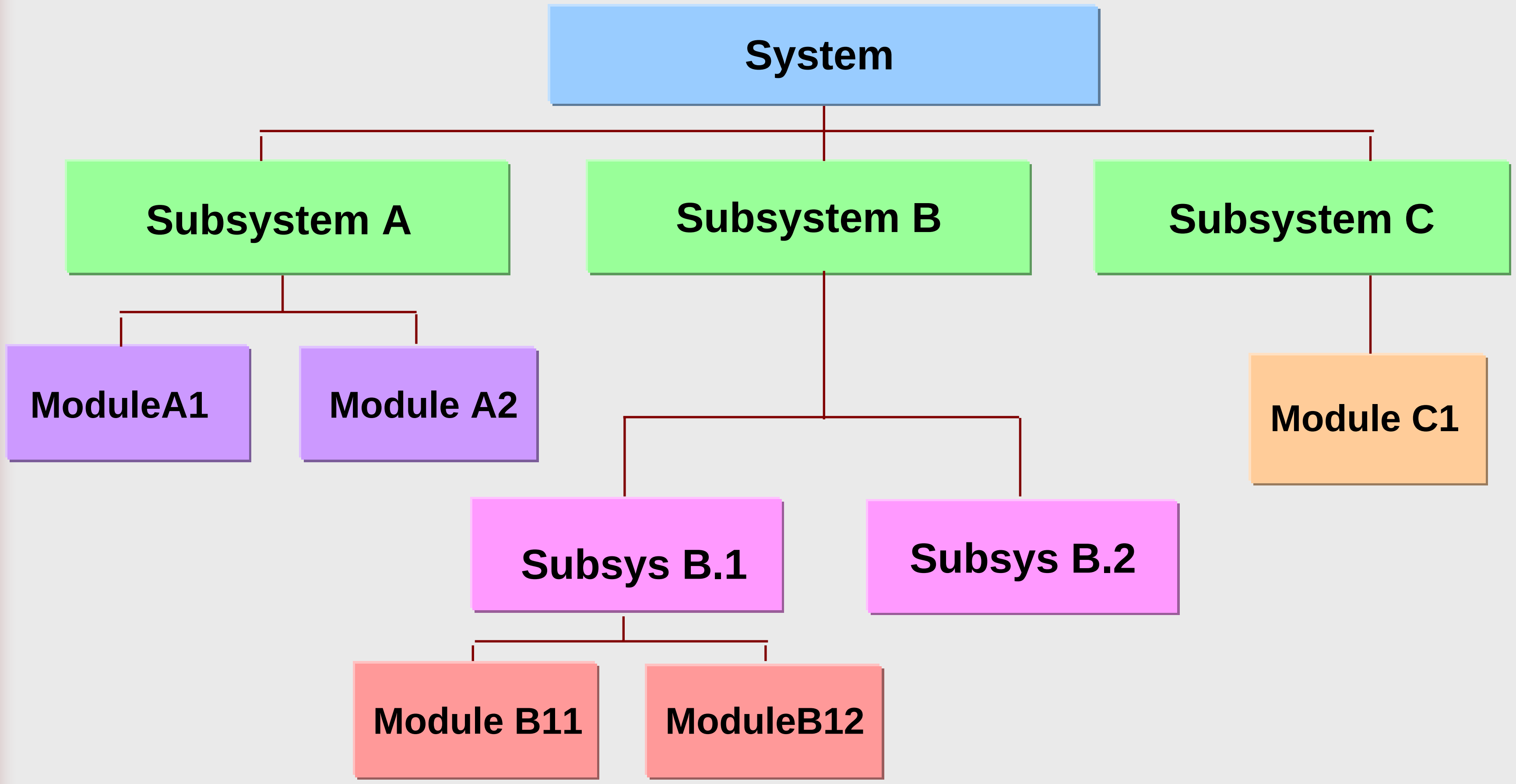
etc...

V-Model Project Development Lifecycle

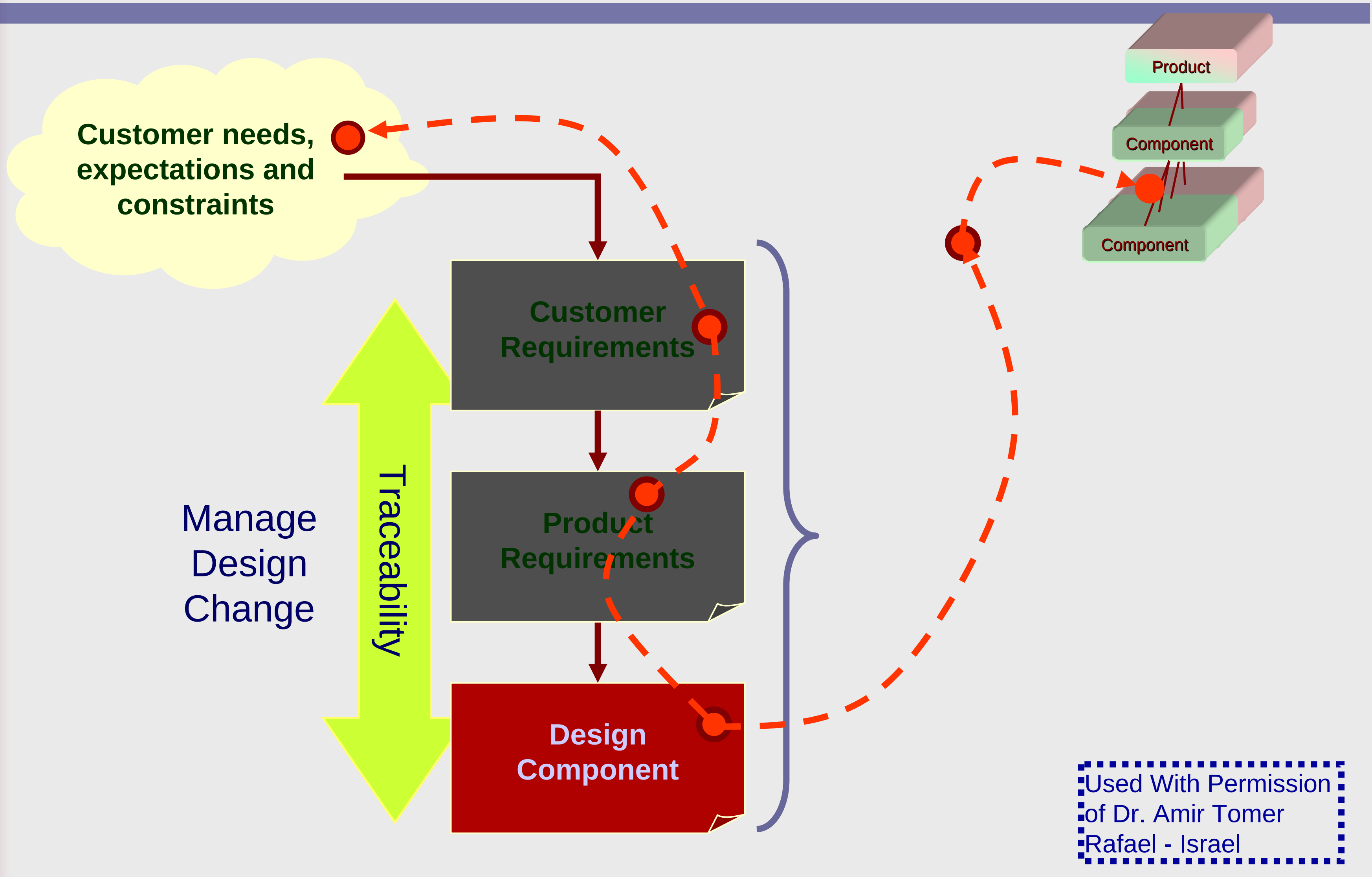


Product Structure

A product system is composed of subsystems which are in turn composed of sub-subsystems, which are composed of modules, which are composed of product components

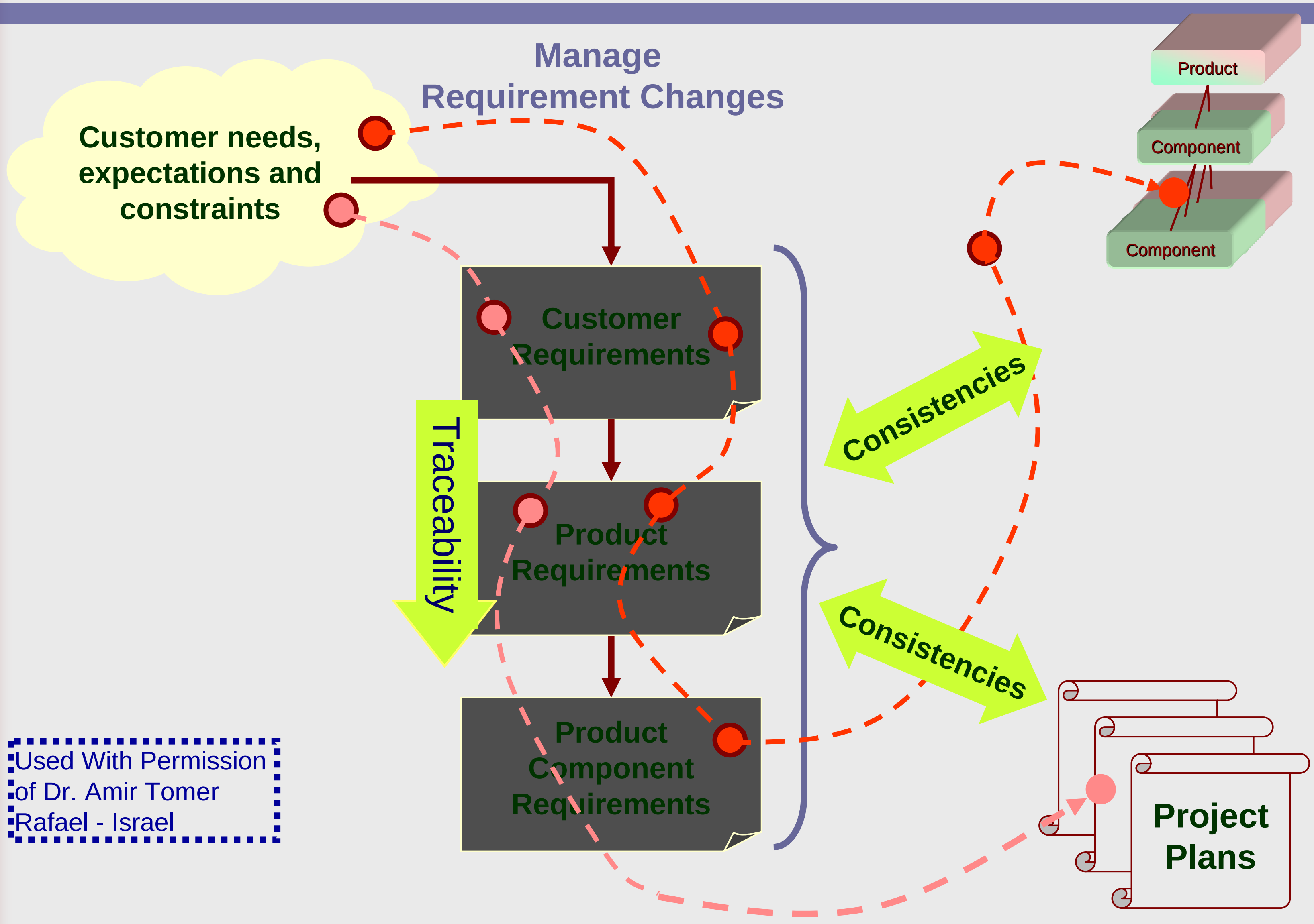


Bi-Directional Traceability



Used With Permission of Dr. Amir Tomer Rafael - Israel

Requirement Traceability

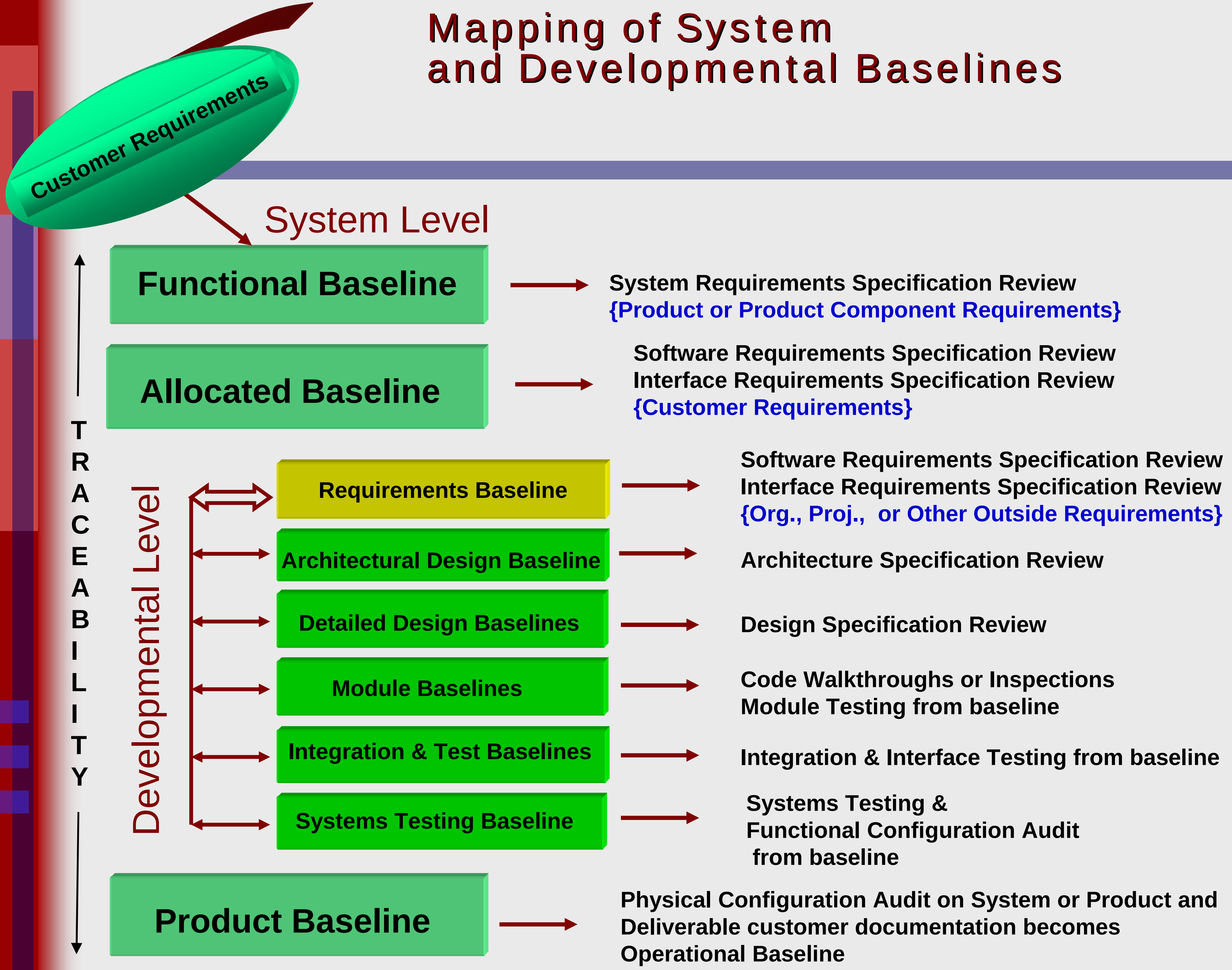


Used With Permission
of Dr. Amir Tomer
Rafael - Israel

Baselining

- Baseline - A set of specifications or work products that has been formally reviewed and agreed on which thereafter serves as the basis for further development, and which can be changed only through change control procedures
- A baseline is an approved snapshot of the one or more configuration items at appropriate points in the development lifecycle
  - Record of a contract
  - Serves as the basis for further development
  - Can be changed only through an agreed upon change control procedure
- A baseline could be
  - A specification (i.e., requirements specification, design specification)
  - A product that has been formally reviewed and agreed upon
  - A partial system

Mapping of System and Developmental Baselines

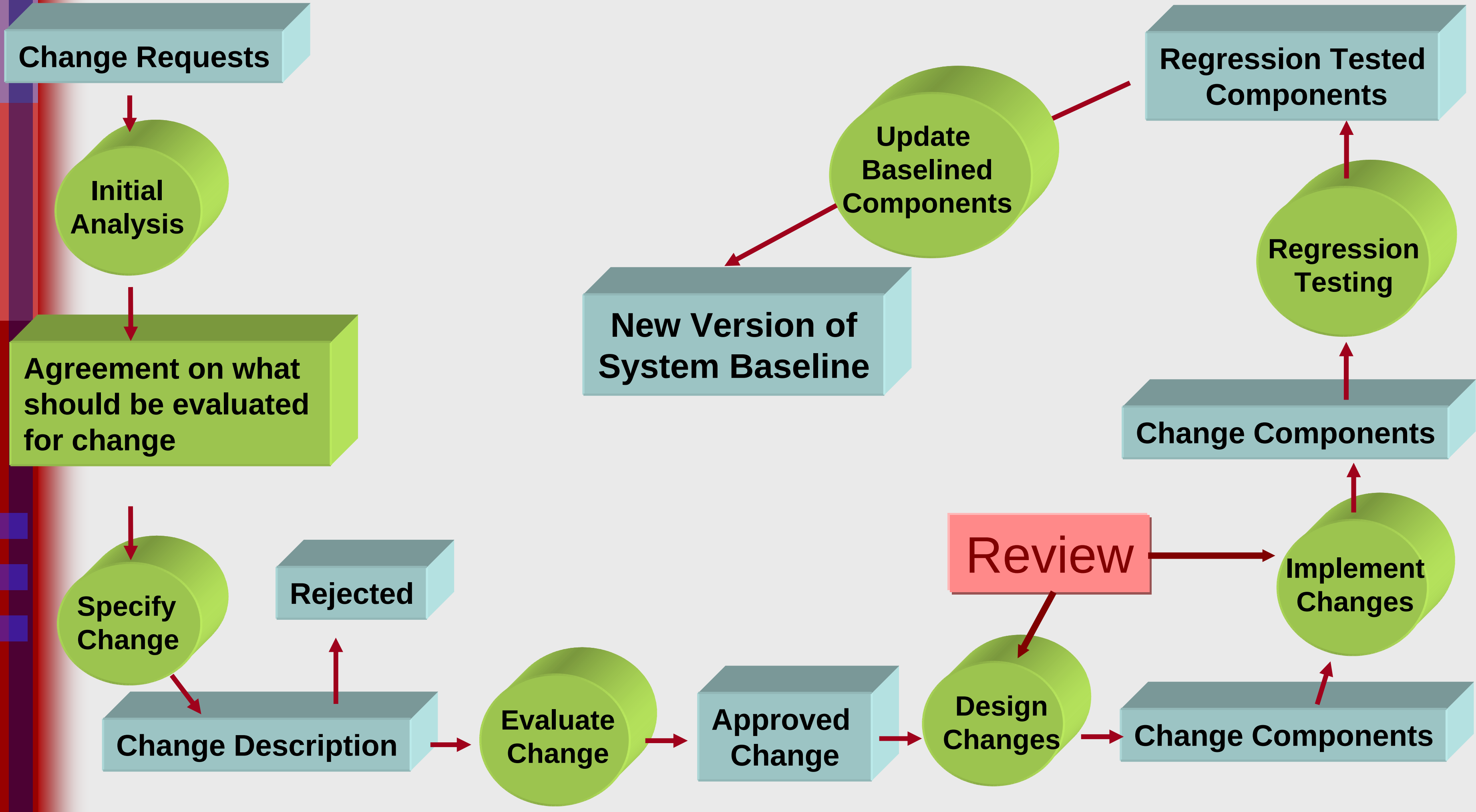


Functional Baseline	System Requirements Specification Review {Product or Product Component Requirements}
Allocated Baseline	Software Requirements Specification Review Interface Requirements Specification Review {Customer Requirements}
Requirements Baseline	Software Requirements Specification Review Interface Requirements Specification Review {Org., Proj., or Other Outside Requirements}
Architectural Design Baseline	Architecture Specification Review
Detailed Design Baselines	Design Specification Review
Module Baselines	Code Walkthroughs or Inspections Module Testing from baseline
Integration & Test Baselines	Integration & Interface Testing from baseline
Systems Testing Baseline	Systems Testing & Functional Configuration Audit from baseline
Product Baseline	Physical Configuration Audit on System or Product and Deliverable customer documentation becomes Operational Baseline

Change Control

- Establishing a change control process that specifies:
  - Who can initiate the change request
  - The individuals, group, or groups who are responsible for evaluating, accepting, and tracking the change proposals for the various baselined products
  - The “change impact” analysis expected for each requested change
  - How the change history should be kept
- Peer Reviews, Unit Testing and regression testing should be performed to ensure that changes have not caused unintended effects on the baselines

Incorporating a Change Into the Current Configuration Baseline



Configuration Control Board (CCB) Function (Organizational Level)

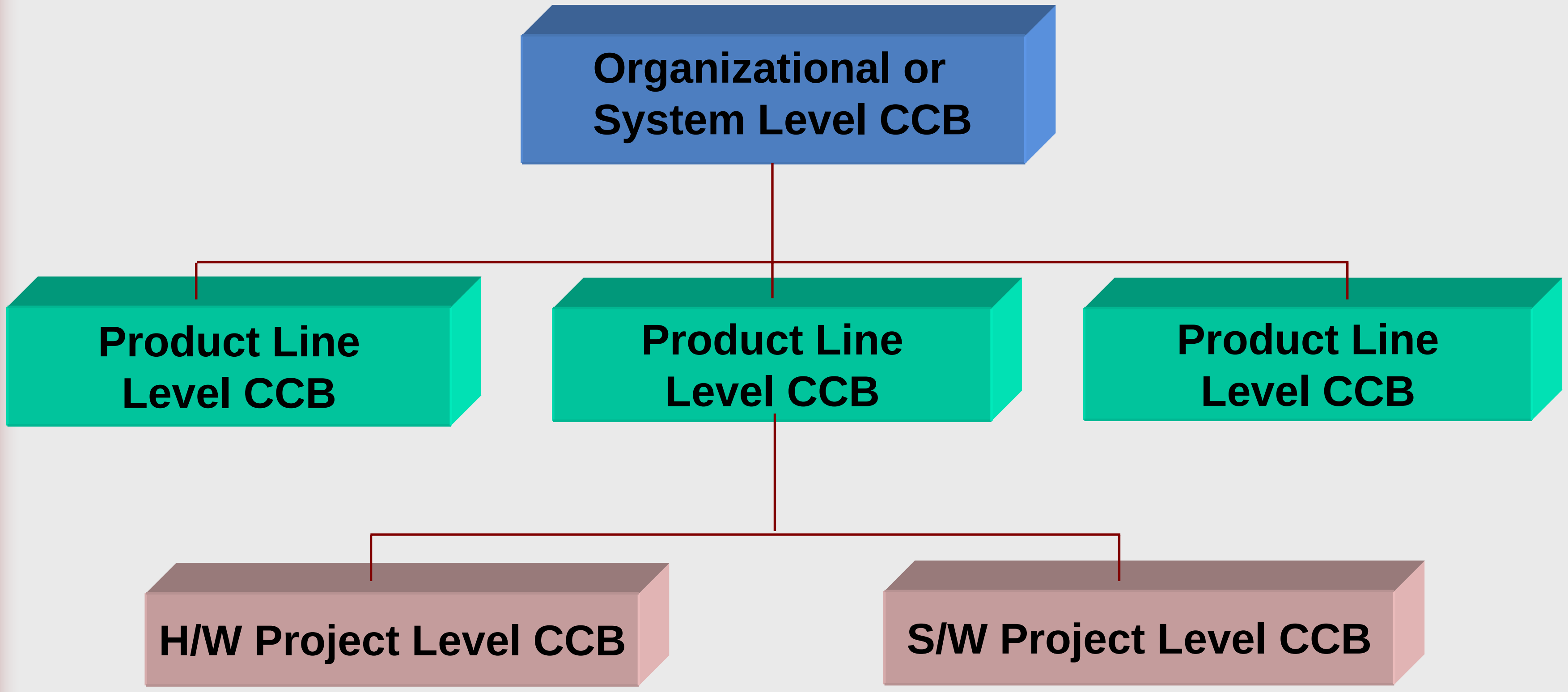
- Authorizes the establishment of baselines and the identification of configuration items
- Represents the interests of all groups who may be affected by changes to the baselines
- Authorizes changes to the baselines
- Authorizes creation and release of products

Organizational CCB Membership

- The Organizational Configuration Control Board (CCB) may include members from:
  - Program Management
  - Systems Engineering
  - Software Engineering
  - Software Quality Assurance
  - Software Configuration Management
  - Independent Test
  - Documentation
  - Hardware Engineering
  - Customer Representative

Hierarchies of CCBs

- Depending on the size of the organization, the number and size of the projects and the complexity of what must be controlled, several CCBs might be needed



Configuration Management System

- The *Configuration Management System* section of the CM Plan describes:
  - Storage Media
  - Procedures for accessing the CMS
  - Tools for accessing the CMS

Configuration Management System (CMS)

- The Configuration Management System
  - Stores and retrieves the configuration items created during the product lifecycle or references to them and prevents unauthorized changes to the baselined items
  - Shares and transfers configuration items between control levels within the CMS
  - Store and recover archived versions of configuration items
  - Store, update, and retrieve configuration management records
  - Create configuration management status reports
  - Preserve the contents of the CMS
    - Backups and restoration of configuration management files
    - Archiving of configuration management files
    - Recovery from configuration management errors

Configuration Management System (CMS) - 2

- Prevents unauthorized access to baselined configuration items
- Supports configuration audits

- The Configuration Management System is a repository where changes to baselines and releases of products and product components take place in a controlled and approved fashion

Interface Control

- Describes which interfaces must be defined and controlled by the project including
  - Organizational interfaces
    - individual
    - project
    - Customer
  - Technical interfaces
    - system
    - life-cycle phase
    - user
    - software
    - hardware
    - communication

Supplier Control

- Ensures that the supplier is able to maintain the integrity of the product or product component it has contracted for, including:
  - Placing necessary life-cycle products under configuration control to ensure consistency with the main development effort
  - Maintaining a supplier configuration management system that will release the agreed upon configuration items of the product or product components to the contracting organization

Supplier Control - 2

- The *Supplier Control* section of the CM Plan should describe:
  - What processes is the Supplier required to follow?
  - What life-cycle work products must be placed under configuration control by the Supplier to ensure consistency with the main development effort?
  - What items are to be supplied by the Supplier including the production of a CM plan?
  - What level of configuration management is required to be supported by the Supplier’s organization?
  - What level of support is required of the Supplier’s configuration management system?

Configuration Management Status Accounting

- Maintaining a continuous record of the status and history of all baselined items and proposed changes to them
- Reports on the traceability of all changes to the baseline throughout the product lifecycle
- Answers the questions
  - What changes have been made to the system?
  - What changes remain to be implemented?

Configuration Auditing

- Configuration audit verifies that the product or product component is built according to the requirements, standards, or contractual agreement
- Verifies that all products or product components have been produced, correctly identified and described, and that all change requests have been resolved

Release

- Basic Release activities to be performed such as:
  - Development of the Version Description Document to describe the product that is being delivered accurately along with any other necessary Release Notes
  - Ensuring that the Product Package is complete and accurate before it is sent to the customer
  - Ensuring that the FCA results show the product about to be shipped meets the requirements and approved requirements change and nothing more
  - Ensuring maintenance documentation is accurate

Release - 2

- Ensuring the installation instructions are accurate
- Ensuring the operator’s manual is accurate and helpful
- Ensuring each customer’s constraints are known before delivery
- Ensuring the customer or end user’s site is prepared to receive and accept delivery
- Ensuring that acceptance testing support is available in the event of problems are discovered during acceptance testing

Release serves as the back end function looking at the quality of the system as it evolves

Configuration Management Status Accounting

Configuration Management Status Accounting

- *Configuration Management Status Accounting* describes:
  - What configuration information should be reported to the project?
    - kinds of reports needed to support the integration of modules or hardware components
    - kinds of reports needed to trace source errors to completion
  - What format should it be placed in?
  - Frequency of reports the project will receive?
  - What procedure should be followed to request configuration information from the configuration management system?

Status Accounting Uses

- Configuration Management Status Accounting provides visibility into the system evolution by recording and reporting the status of all configuration items and the status of all requests for change

Status Accounting Uses - 2

- Questions that Configuration Management Status Accounting should be able to answer include:
  - What is the status of an item?
    - A developer may want to know whether a specification has been fully approved
    - A developer may want to know whether a subsystem has been tested so that the programmer can test his modules which interfaces with that subsystem
    - A project leader will wish to track the progress of a project as items are developed, reviewed, tested and integrated

Status Accounting Uses - 3

- Has a change request been approved or rejected by the CCB?
  - The originator of a change request will want to know if the CCB has approved or rejected the request
- Which version of an item implements an approved change request?
  - Once a requested change of a configuration item is implemented, the originator and other developers will want to know which version of the configuration item contains the change

Status Accounting Uses - 4

- What is different about a new version of a system?
  - A new version of a product component should be accompanied by a document listing the changes from the previous version
    - The difference between successive baselines should be able to be clearly described
  - The change list should include both enhancements and fixes to faults
  - Any faults that have not been fixed should also be described

Status Accounting Uses - 5

- What is the root cause for a high number of changes to a work product?
  - Number of change requests at the organizational level starting with requirements change requests?
  - Number of work products that have change requests at the project level by the developers, project managers or other support personnel
  - Changes required by the detection of faults

Status Accounting Uses - 6

- How many faults are detected each month and how many are fixed?
  - Faults are continuously detected during the operational use of the system
  - Comparing the number of detected and fixed faults helps to assess the stability of the latest release of the system
  - Tracking the number of faults and fixes also helps the Program Manager to decide when to make a new release of the system

Status Accounting Uses - 7

- What is the cause of the trouble report?
  - Trouble reports can be categorized by their causes
    - violation of programming standards
    - inadequate user interface
    - left out customer requirements
  - Sometimes when it is discovered that many faults have a similar cause, action can be taken to improve the process and stop such faults from recurring

Configuration Auditing

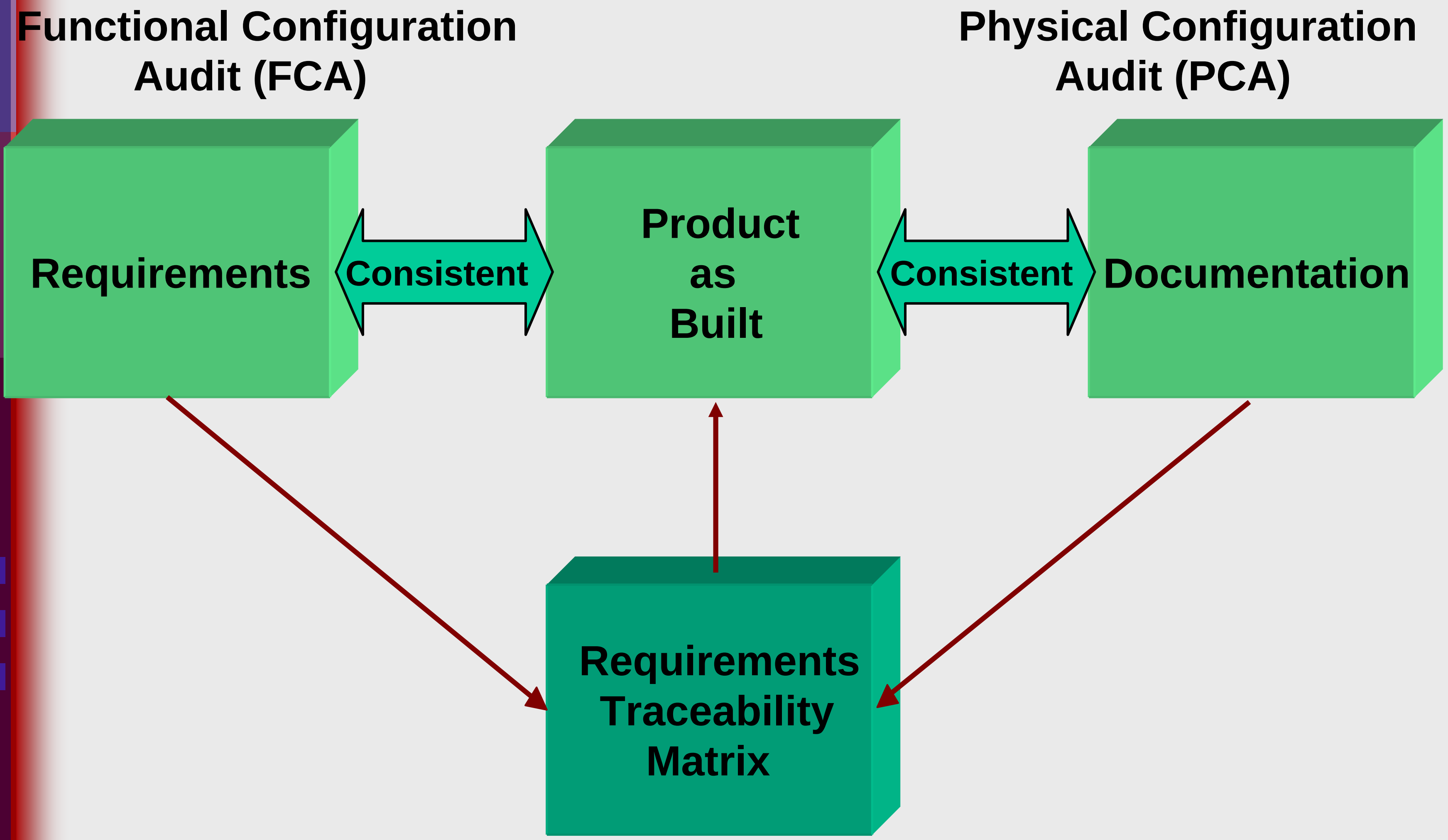
Configuration Auditing

- Configuration auditing verifies that the product is built according to the requirements, standards, or contractual agreement
- Verifies that all product components have been produced, correctly identified and described, and that all change requests have been resolved

Configuration Auditing - 2

- Baseline audits should be conducted throughout the project lifecycle at the end of a phase, before a designated change to the baseline, or perhaps before or after a milestone
  - The integrity of the baselines are assessed
  - The completeness and correctness of the baseline library contents are verified
    - based on the requirements as stated in the plan and the approved requirements change requests
  - The product's functionality and performance are compared to the requirements
  - The documentation that is baselined for maintenance activities and for operational use is compared to the requirements

FCA and PCA Audits


Functional Configuration Audit (FCA)
Physical Configuration Audit (PCA)
Requirements
Consistent
Product as Built
Consistent
Documentation
Requirements Traceability Matrix

Functional Configuration Audit

- The objective of the Functional Configuration Audit (FCA) is to provide an independent evaluation of configuration item, verifying that each configuration item's actual functionality and performance is consistent with the Requirements Specification
- Normally the FCA is conducted prior to the product delivery to verify that all requirements specified in the Requirements Specification have been met

Project Feedback From a Functional Configuration Audit

- The Project Leader can expect the feedback from the FCA to contain information similar to the following:
  - How the formal test documentation compared against the test data
  - The accuracy of the verification and validation reports
  - A statement indicating if all approved changes were incorporated and verified
  - If updates to previously delivered documents were accurate and consistent
  - If the code addresses all and only the documented requirements (technical, non-technical, quality)

Project Feedback From a Functional Configuration Audit - 2

- If all testing had been accomplished with appropriate test documentation and validated data to establish configuration item performance
- If all performance parameters were verified completely through testing, simulation or other analyses to ensure that the configuration item met the established performance criteria
- Manuals conform with the systems and functional descriptions
  - User's manuals
  - Maintenance manuals
  - Operator's manuals

Physical Configuration Audit

- The objective of the Physical Configuration Audit (PCA) is to provide an independent evaluation of the system configuration items to confirm that each CI that makes up the “as built” system maps to its specifications
- This audit must be held to verify that the product and its documentation are internally consistent and are ready for delivery

Project Feedback From a Physical Configuration Audit

- The Project Leader can expect the feedback from the PCA to contain information similar to the following:
  - An indication if the the system specification was complete or not
  - An audit of the Functional Configuration Audit report for discrepancies and actions taken
  - An architectural design consistent with the detailed design components
  - Components built from the design documents
  - Module listings compliant with the approved coding standards

Results of Configuration Management

What CM Provides

- A key role of Configuration Management is to control changes actively in order to answer the following questions:
  - What is the current product or product component configuration ?
  - What is the status of my product component?
  - What changes have been made to my product components?
  - Does anyone else’s changes affect my product components?

What CM Provides - 2

- CM provides visibility into the status of the evolving product or product component
- CM answers the Who, What, When, and Why
  - Who made the changes?
  - What changes were made to the product or product component?
  - When were the changes made?
  - Why were the changes made?

Project Leader Support From CM

- A strong understanding and implementation of CM helps the Project Leader
  - Control changes to the requirements
  - Allow the project members to develop at a fast pace without interference during the early stages of development
  - Control developers “improving” the product component (code for software) during when it is at the infamous 90% complete stage
- Assists the Project Leader to develop in an iterative approach thereby reducing complexity and risk

Project Leader Support From CM - 2

- Assists the Project Leader in producing accurate and up to date Status Reports:
  - Provides status reports to the Project Leader indicating what configuration items are undergoing the most change in terms of number of changes and frequency of changes
  - Provides traceability to give the Project Leader a level of confidence that what the developers are developing is what is demanded by the requirements and nothing more
  - Helps ensure the integrity and consistency of the evolving system so that the product or product component and associated documentation and specifications are synchronized

Summary: The CM Functions

What is the system configuration?	Identification	The system consists of the following baseline documents and products:...
How are changes to the configuration controlled?	Control	The steps to process changes are...
What changes have been made to the system?	Status Accounting	The system configuration and related changes at this line are the combination of the following baselines, changes, pending changes:...
Does the system satisfy the requirements?	Auditing	The system as currently built differs from the baselines and approved changes as follows:...

Kasse Initiatives Contact Information

United States Address

Tim Kasse

CEO & Principal Consultant

Kasse Initiatives LLC

PMB 293

1900 Preston Road # 267

Plano, Texas 75093

United States of America

+1 972 – 987 – 7606 Business

+1 972 – 987 – 7607 FAX

kassetc@aol.com

www.kasseinitiatives.com

Tim Kasse

- CEO and Principal Consultant of Kasse Initiatives
- Visiting Scientist - Software Engineering Institute
- Visiting Fellow - Institute for Systems Science / National University of Singapore
- Author of Action Focused Assessment for Software Process Improvement
- Author of Practical Insight Into CMMI

Kasse Initiatives Contact Information

Pamelia S. Rost
Executive VP
Business Development

Kasse Initiatives LLC

PMB 293

1900 Preston Road # 267

Plano, Texas 75093

United States of America

+1 972 – 987 – 9878 Business

+1 972 – 987 – 7607 FAX

pamelia.rost@kasseinitiatives.com

www.kasseinitiatives.com

Kobi Vider

Chief Methodologist

- Principal Consultant
- Coordinate the integration and synchronization of newly developed methodology documents with existing methodologies
- Assess the applicability of industry state-of-the-art best practices related to software development methodologies, techniques, standards, and tools
- Six Sigma, Lean, SCRUM, Agile

KobiVP@aol.com

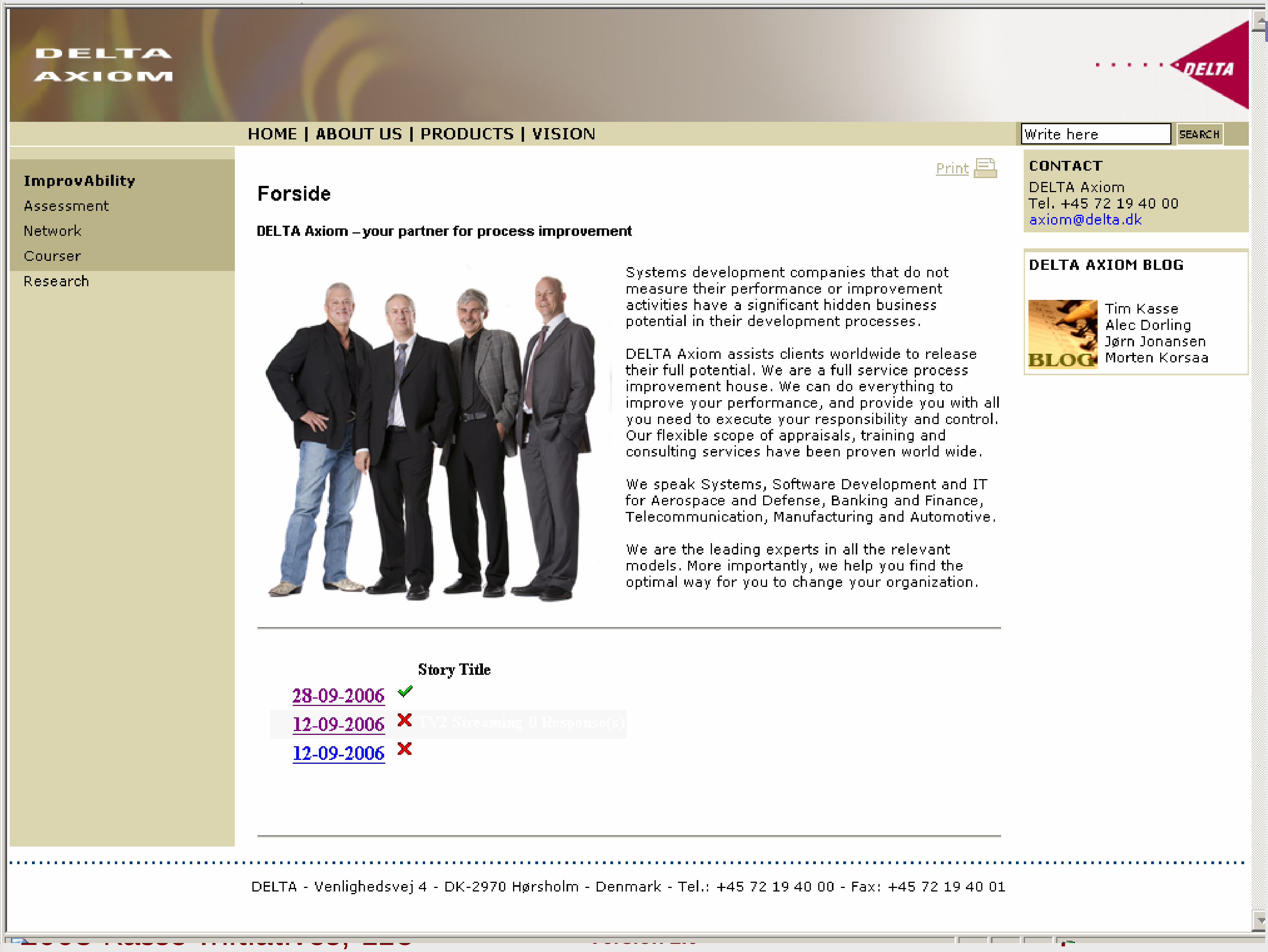
Ahmet Betil

- CEO and Principal Consultant of WYNNTRYST LLC
- Process Improvement Consultant
- E-Learning Facillator
- KI Instructor

Wynntryst LLC
26 Baycrest Drive South
Burlington, VT 05403, United States
Phone: (802) 578-1593
www.wynntryst.com

WYNNTRYST LLC

Delta Axiom



HOME | ABOUT US | PRODUCTS | VISION

ImprovAbility
Assessment
Network
Courser
Research

Forside

DELTA Axiom – your partner for process improvement

Systems development companies that do not measure their performance or improvement activities have a significant hidden business potential in their development processes.

DELTA Axiom assists clients worldwide to release their full potential. We are a full service process improvement house. We can do everything to improve your performance, and provide you with all you need to execute your responsibility and control. Our flexible scope of appraisals, training and consulting services have been proven world wide.

We speak Systems, Software Development and IT for Aerospace and Defense, Banking and Finance, Telecommunication, Manufacturing and Automotive.

We are the leading experts in all the relevant models. More importantly, we help you find the optimal way for you to change your organization.

		Story Title
28-09-2006	✓	
12-09-2006	✗	TV2 Streaming 0 Response(s)
12-09-2006	✗	

CONTACT
DELTA Axiom
Tel. +45 72 19 40 00
axiom@delta.dk

DELTA AXIOM BLOG
BLOG
Tim Kasse
Alec Dorling
Jørn Jonansen
Morten Korsaa

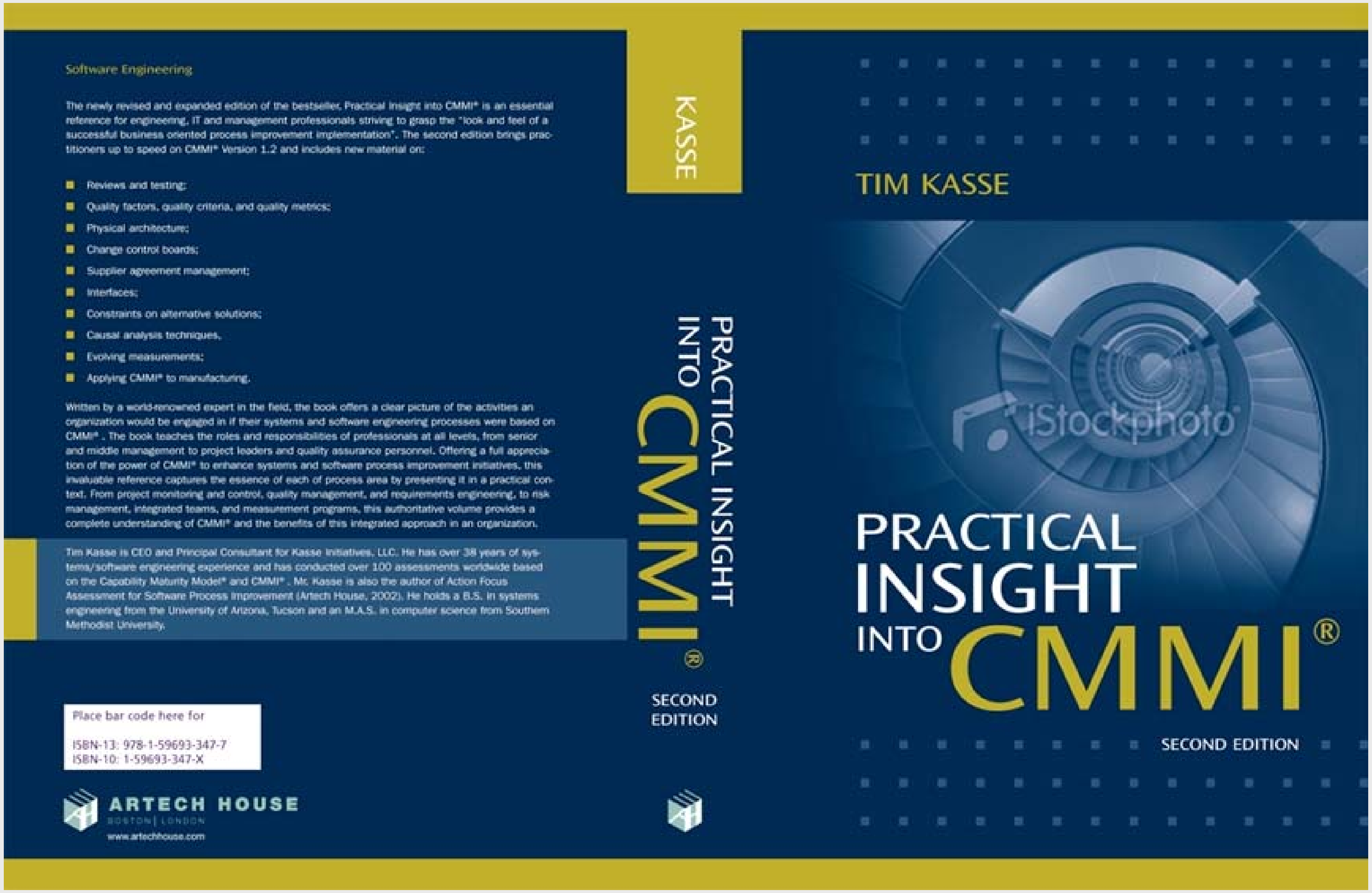
DELTA - Venlighedsvej 4 - DK-2970 Hørsholm - Denmark - Tel.: +45 72 19 40 00 - Fax: +45 72 19 40 01

© 2006 Kasse Initiatives, LLC

KI Cooperating Partners

- Delta Axiom
- Wynntryst
- Process Plus Inc.
- Software Inc.
- SPI Partners
- KT Bits
- Q:PIT
- KVP Consulting

Books
From Kasse Initiatives



Software Engineering

The newly revised and expanded edition of the bestseller, Practical Insight into CMMI® is an essential reference for engineering, IT and management professionals striving to grasp the "look and feel of a successful business oriented process improvement implementation". The second edition brings practitioners up to speed on CMMI® Version 1.2 and includes new material on:

- Reviews and testing;
- Quality factors, quality criteria, and quality metrics;
- Physical architecture;
- Change control boards;
- Supplier agreement management;
- Interfaces;
- Constraints on alternative solutions;
- Causal analysis techniques,
- Evolving measurements;
- Applying CMMI® to manufacturing.

Written by a world-renowned expert in the field, the book offers a clear picture of the activities an organization would be engagegd in if their systems and software engineering processes were based on CMMI® . The book teaches the roles and responsibilities of professionals at all levels, from senior and middle management to project leaders and quality assurance personnel. Offering a full appreciation of the power of CMMI® to enhance systems and software process improvement initiatives, this invaluable reference captures the essence of each of process area by presenting it in a practical context. From project monitoring and control, quality management, and requirements engineering, to risk management, integrated teams, and measurement programs, this authoritative volume provides a complete understanding of CMMI® and the benefits of this integrated approach in an organization.

Tim Kasse is CEO and Principal Consultant for Kasse Initiatives, LLC. He has over 38 years of systems/software engineering experience and has conducted over 100 assessments worldwide based on the Capability Maturity Model® and CMMI® . Mr. Kasse is also the author of Action Focus Assessment for Software Process Improvement (Artech House, 2002). He holds a B.S. in systems engineering from the University of Arizona, Tucson and an M.A.S. in computer science from Southern Methodist University.

Place bar code here for

ISBN-13: 978-1-59693-347-7
ISBN-10: 1-59693-347-X

ARTECH HOUSE
BOSTON | LONDON
www.artechhouse.com

KASSE

PRACTICAL INSIGHT INTO CMMI®

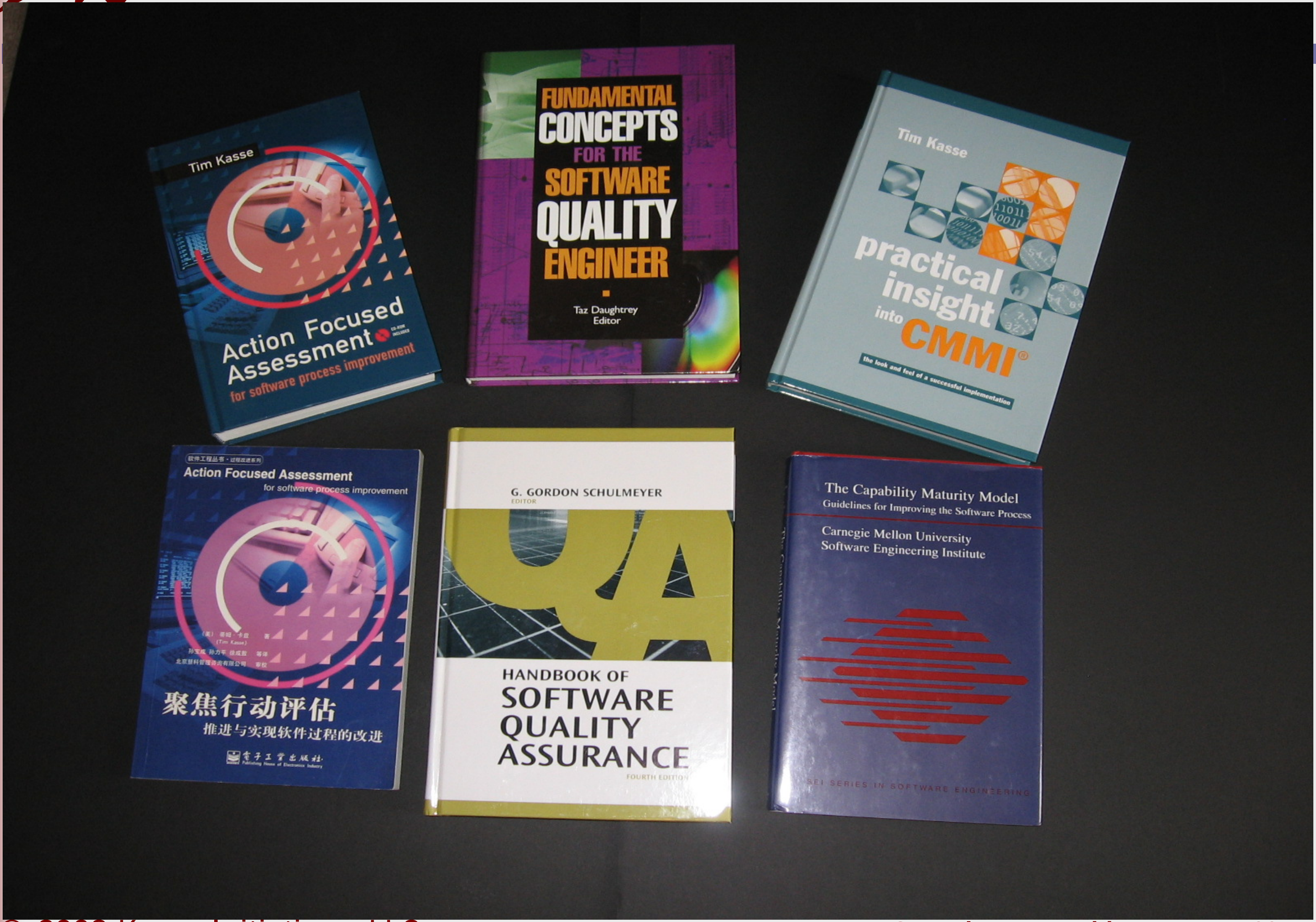
SECOND EDITION

TIM KASSE

PRACTICAL INSIGHT INTO CMMI®

SECOND EDITION

Books From Kasse Initiatives - 2



Tim Kasse
Action Focused Assessment
for software process improvement
FUNDAMENTAL CONCEPTS FOR THE SOFTWARE QUALITY ENGINEER
Taz Daughtrey Editor
Tim Kasse
practical insight into CMMI®
the look and feel of a successful implementation
Action Focused Assessment
for software process improvement
聚焦行动评估
推进与实现软件过程的改进
G. GORDON SCHULMEYER EDITOR
HANDBOOK OF SOFTWARE QUALITY ASSURANCE
FOURTH EDITION
The Capability Maturity Model
Guidelines for Improving the Software Process
Carnegie Mellon University Software Engineering Institute
SEI SERIES IN SOFTWARE ENGINEERING

Books From Kasse Initiatives - 3

- Tim Kasse, Practical Insight to the CMMI – 2nd Edition, Artech House, Cambridge, Massachusetts, 2008
- Schulmeyer, G. Gordon, *Handbook of* Software *Quality Assurance (4th Ed)* – Artech House, 2007. Tim Kasse wrote the Chapter "CMMI PPQA Relationship to SQA"
- Tim Kasse, Practical Insight to the CMMI, Artech House, Cambridge, Massachusetts, 2004
- Tim Kasse, Action Focused Assessment for Software Process Improvement, Artech House, Cambridge Massachusetts, 2002.

Books
From Kasse Initiatives - 4

- Tim Kasse, Action Focused Assessment for Software Process Improvement, (Chinese Translation). Electronics Industry , Beijing, China, 2002.
- Taz Daughtrey, Fundamental Concepts for the Software Quality Engineer, ASQ Quality Press, 2002. Tim Kasse and Dr. Pat McQuaid contributed the chapter on Software Configuration Management for Project Leaders.
- Mark C. Paulk, Charles V. Weber, Bill Curtis, Mary Beth Chrissis, The Capability Maturity Model - Guidelines for Improving the Software Process, Carnegie Mellon University Software Engineering Institute, Pittsburgh, Pennsylvania, 1995. Tim Kasse is listed as a contributor