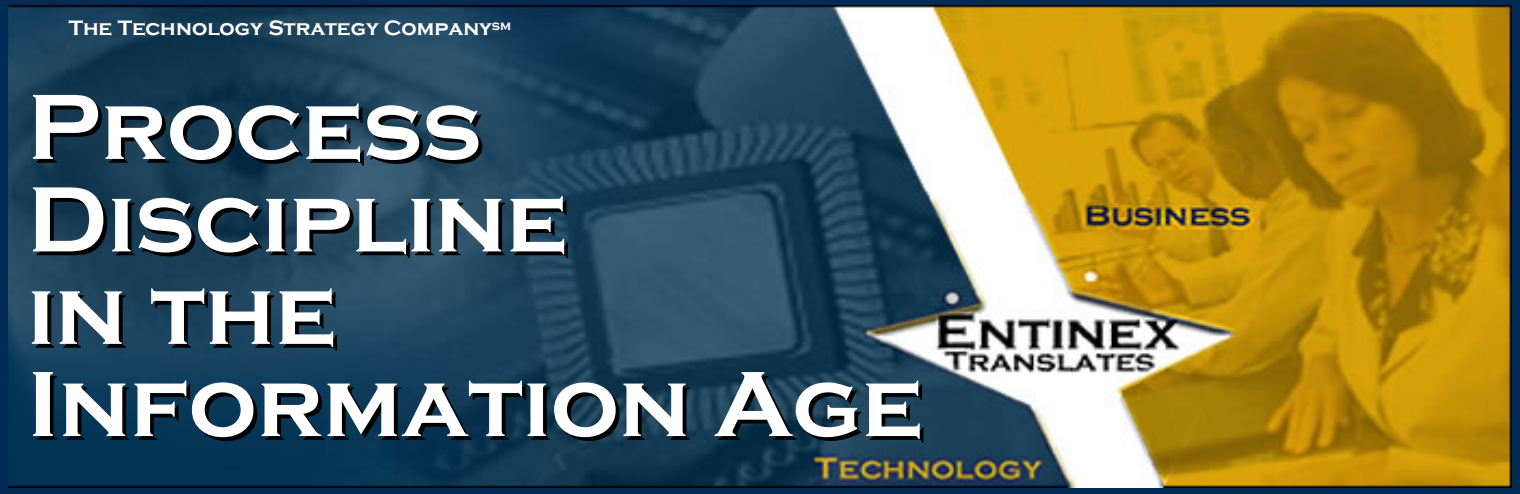
THE TECHNOLOGY STRATEGY COMPANY℠

PROCESS DISCIPLINE IN THE INFORMATION AGE



Rethink the "Quality" Abstraction

Hillel Glazer
Entinex, Inc.
spi@entinex.com
301. 384. 4203

Chris Hagenbuch, PMP, CPSM
InfoZen
chagenbuch@infozen.com
301.980.0602

Purpose of Presentation

- To shed light on…
  - Today's software market & how it's different from the software market when QA (process discipline) was first applied.
  - Why traditional QA fails in today's software development environment.
  - How QA needs to be structured to work in today's software development environment.
  - What "agile" software development is and isn't.
  - How agile software development can be disciplined.

Agenda

- Introduction
- Understanding Agile
- Role and Goal of "QA"
- Historical Role of QA
- QA in Business
- A Word About Development Processes
- QA in the Context of Development Processes
- Rethinking the Quality Abstraction
- An Implementation Example with Scrum
- Conclusion

Who Is This Guy?

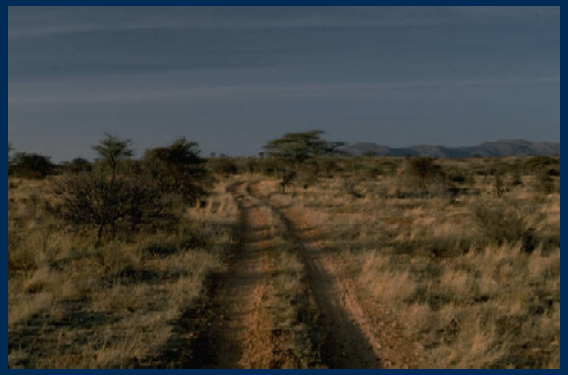
- Personal Introduction
- Fresh Fish in the Fire
- Jedi training
- The Dark Side
- Why Talk?
- Staying/Straying on Topic
- "…Slings and Arrows…"
- Tom Peters!

Agenda

- Introduction
- Understanding Agile
- Role and Goal of "QA"
- Historical Role of QA
- QA in Business
- A Word About Development Processes
- QA in the Context of Development Processes
- Rethinking the Quality Abstraction
- An Implementation Example with Scrum
- Conclusion

Introduction

- Why "Re-Think" the Quality Abstraction?
- QA's Legacy Mindset
- Software Today and Yesterday
- Movement in the Software Industry



- Why/Where Lightweight and Heavyweight Collide
- Disciplined vs. Undisciplined
- QA as a Valuable Asset

Why "Re-Think" The Quality Abstraction?

- Align process and product technologies
- Align development environment
- Align with market forces

QA's Legacy Mindset

- Large Products
- Software a component of the product
- Technology trade-off
- Software is now the entire product

Software Today and Yesterday

- Compare today's software products to those 10, 15, 20+ years ago
- How are they similar?
- They are more different than they are similar.
- QA hasn't changed with the technologies and methodologies

Movement in the Software Industry

- Companies attempting process QA initiatives found everywhere on the side of the road
- Agile/Lightweight seen as a "way out"
- Cannot ignore the trend
- "Lightweight" in response to "Heavyweight"

Why/Where Light & Heavy Collide

- Attitude
- Strengths
- Weaknesses
- Typical approach to QA propagates legacy methods & mindset
- There's no such thing as robust QA in lightweight development ?!

Disciplined vs. Undisciplined

- Lightweight is thought of as undisciplined?
- Could it be QA's fault?
- Can lightweight/agile development also be robust?
- Can QA become appropriately agile?
- Can a mindset "re-set" about QA be applied?
- Could an abstraction be created for QA that works in any environment?
- Could it be used to improve QA in non-lightweight software environments?

QA as a Valuable Asset

- QA activities are expected to be:
  - value-added
  - a component of a comprehensive product development process.
- Look at QA in terms of:
  - its basic goals.
  - how to adapt what QA professionals do to meet those goals.
- Developed a QA approach that:
  - works in any environment
  - Is still in complete compliance with standards, and policies.
- A change in abstraction will cause:
  - how QA "shows up" on a project,
  - ***not*** what QA is expected to accomplish.

Agenda

- Introduction
- Understanding Agile
- Role and Goal of "QA"
- Historical Role of QA
- QA in Business
- A Word About Development Processes
- QA in the Context of Development Processes
- Rethinking the Quality Abstraction
- An Implementation Example with Scrum
- Conclusion

Understanding "Lightweight" / "Agile"

- Lightweight Reputation
- True Purpose of Lightweight/Agile Development
- Development in the Absence of Process?
- Working Definition of Lightweight/Agile Development
- Agile Alliance

Lightweight Reputation

- Lightweight development varies widely from organization to organization.
- Lightweight/Agile the reputation as undisciplined.
- Narrow implementations of the concepts, rarely following any formal development guidelines.
  - ∴ the reputation is unfair.
- Coding without rules, process discipline, or management tools is undisciplined.
  - This is not what agile development is.
  - Any more than the original intent of effective QA was to be heavy-handed.

True Purpose of Agile Development

- The purpose of lightweight development is to allow for better productivity.
- The enemy of productivity is heavy-handed process controls.
- True, some developers pursue lightweight development thinking they can shed controls, checks, and balances necessary to make good products.
- This is far from what lightweight is about.

Development in the Absence of Process?

- If this were true, then lightweight developers would be operating under a modus operandi that reads:
  - "produce quality software in the absence of any process".
    - This would be absurd.
    - Lightweight supporters do not agree with this.

- It's not the absence of process that makes a development method lightweight.
- It's the absence of unnecessary or obstructive processes that makes a method lightweight.

Working Definition of Agile Development

- The minimum, most unobtrusive approach to developing software that produces a quality product when the customer expects to get it, at the price they expect to pay.

Agile Alliance

- Principles:
  - **Satisfy the Customer** thru valuable software
  - **Changes happen**, harness them for the customer's benefit
  - Deliver **working product frequently**
  - The **business must work with the developers**
  - Hire **motivated people, support them, let them work**
  - **Face-to-face beats paper**
  - **Working software is the best** measure of progress
  - If it's **not sustainable, it's not agile**
  - Agility depends on **continuous attention to technical excellence & good design**
  - **Simplicity is key** to maximizing work not done
  - **Self-organizing teams** produce the best technical results
  - *Regularly reflect on becoming more effective and tune & adjust.*

Agile Alliance, 2

- Manifesto:

"We value:

*Individuals and interactions **over** processes and tools*

*Working software **over** comprehensive documentation*

*Customer collaboration **over** contract negotiation*

*Responding to change **over** following a plan*

That is, while there is value in the items on the right, we value the items on the left more."

- Is this anti-process?
- Can anyone prefer the process to actually delivering the product?

Agenda

- Introduction
- Understanding Agile
- Role and Goal of "QA"
- Historical Role of QA
- QA in Business
- A Word About Development Processes
- QA in the Context of Development Processes
- Rethinking the Quality Abstraction
- An Implementation Example with Scrum
- Conclusion

Role and Goal of "Quality Assurance"

- Proactive or Reactive?
- How to Entice Developers to Follow a Process
- Working Definition of QA
- Working Policy Statement of QA

Proactive or Reactive?

- Realistically, in the typical QA approach not all activities performed to "satisfy QA requirements"
  - productive,
  - pro-active,
  - value-added contributions to producing the product.
- Otherwise, developers would use typical QA processes.

How to Entice Developers to Follow a Process

- Time away from development isn't productive
  - (documentation of work already performed)
- Reconciling heavyweight and lightweight practices will be found by bridging this gap.
- Create processes that parallel development.

Working Definition of QA (Process Discipline)

- *A process/effort that ensures that processes are followed, that*
- *the processes have us doing*
  - *the right things,*
  - *the right way, and*
- *when they fail to be used or fail to perform as expected we have a way to*
  - *correct,*
  - *adjust, or*
  - *escalate the matter until it is resolved to everyone's satisfaction.*

Working Policy Statement of QA

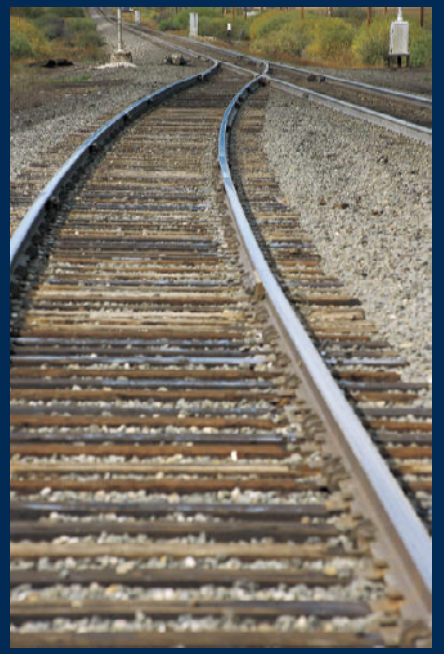
- *All processes must forge a working relationship which*
  - *actively supports development's productive activities*
  - *avoids creating additional effort for development functions outside of the project's stated development function processes*
  - *designs processes in collaboration with the project's development community*
  - *allows the process owners to achieve their process and product oriented objectives*
  - *reaches consensus on a balance between process and productivity.*
- *The goal:*
  - *fully integrate the necessary process steps into activities that add value to the development effort while*
  - *resulting in insight, predictability, measurements and traceability of process effectiveness.*

Agenda

- Introduction
- Understanding Agile
- Role and Goal of "QA"
- Historical Role of QA
- QA in Business
- A Word About Development Processes
- QA in the Context of Development Processes
- Rethinking the Quality Abstraction
- An Implementation Example with Scrum
- Conclusion

Historical Role of QA (Process Discipline)

- QA Has Come a Long Way
- QA Has Far to Go
- QA's Value to Business
- What Fuels Processes?
- Original QA Processes
- Applied to Software?
- One Transformation Matrix After Another



QA (Process Discipline) Has Come a Long Way

- QA undergoes continuous improvement in terms of its application as well as acceptance.
- QA is an official component of many project plans and a valued resource in many projects and organizations.
- QA can hold up a project with process problems, and "dress down" a project manager for skipping steps.

QA (Process Discipline) Has Far to Go

- Unfortunately, QA is still sidelined too often when business needs take priority.
- Too often, QA still has the reputation of "policing" rather than a contributing to the effort.
- Frequently, business owners will bypass managers and go directly to developers when such layers are seen as getting in the way.

QA's Value to Business

- Showing the business value of QA through analyses, 6-sigma SPC, and other techniques are still more ***re***active than ***pro***-active.
- What developers (and executives) want are processes that implement QA so that they don't
  - slow progress,
  - break momentum, or
  - install the sense that people are being policed.
- Such processes are demoralizing.

What Fuels Processes?

- Processes are fueled by people
- People hate heavy-handed processes.
- Developers seek "lightweight" methods in hopes of finding refuge from heavy-handed processes.
- Many throw out the mantle of all processes.
- The origins of QA standards explains much…….

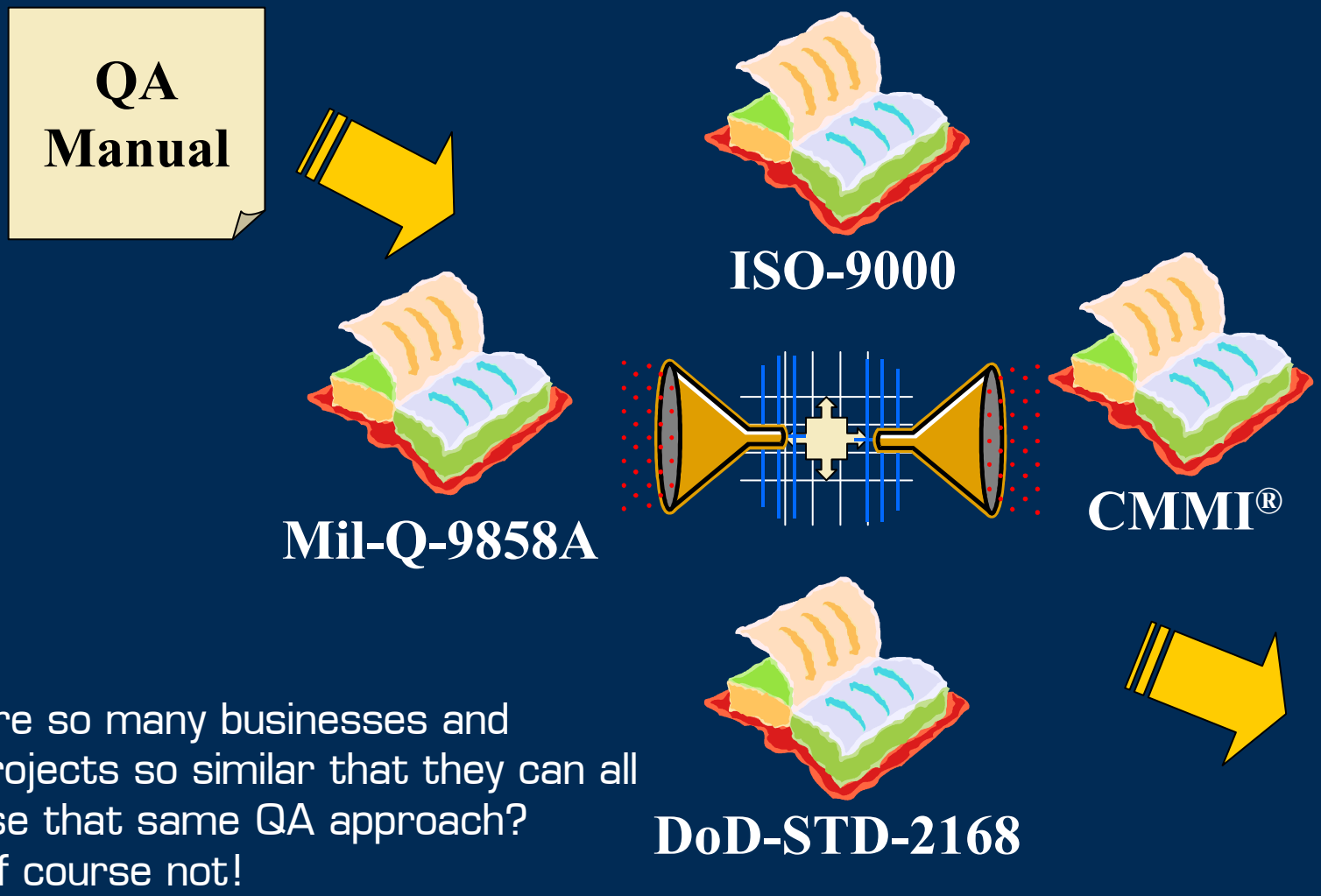
Original QA Processes

- Early software projects were
  - big,
  - slow, and
  - geographically dispersed
- Early projects were characterized by
  - layers of bureaucracy
  - designed around project management methods that also built tanks, planes, and ships.
- Based on manufacturing work-flow and controls.

Applied to Software?

- These QA methods fail to achieve their intended goals.
- *Software development paradigm shares very little with the manufacturing paradigm.*
- Methods of performing QA have not made the shift across the industry.
- Defense and similar large-scale old-style projects shaped much of what is known today about QA.
- Compared to today's technologies and the speed to market, these legacy projects provide a very limiting pool of experience.

One Transformation Matrix After Another



Are so many businesses and projects so similar that they can all use that same QA approach? Of course not!

Agenda

- Introduction
- Understanding Agile
- Role and Goal of "QA"
- Historical Role of QA
- QA in Business
- A Word About Development Processes
- QA in the Context of Development Processes
- Rethinking the Quality Abstraction
- Examples from Everyday Activities
- Conclusion

QA in Business

- Absorption of QA in Larger Projects
- Needs of Development: QA at the Pace of the Project
- Needs of Company: *Working Product*
- QA the "Easy" Way vs. the Way QA Works Best

Absorption of QA in Larger Projects

- The non-productive activities and paperwork created by legacy QA abstractions is most felt at the development level.
- In many large, complex projects, the additional effort and time needed to follow the processes are easily absorbed by the project.
- The pace of these projects are such that the deliberate (if not judicious) addition of time and work can be handled.

Development's Needs: QA at Project's Pace

- Development needs to:
  - stay productive,
  - control costs, and
  - keep people motivated.
- The effort to follow the process should not overshadow the pace of the project.
- Agile development recognizes the need for processes that allow a project to get done at the pace of the project.
- Many processes, QA included, have fallen short because they do not account for the pace and complexity of the project.

Needs of Company: Working Product

- Modern software needs QA processes:
  - more closely fitted to each project,
  - dynamically adapting to the project and making development cheaper, better, and faster on every subsequent project.
- Truly add business value to the QA process.
- On time working product is a must.
- Processes must reflect the demands of the customer. First and foremost.
- Processes must be adaptive and scalable to handle exceptions.

QA the "Easy" Way vs. the Way QA Works Best

- Historically, legacy QA processes not designed with attention to business goals.
- Latest models promote processes that add value.
- Few implementations ever achieve that.
- Instead, companies supplement existing processes with a disruptive, paper-intensive meta-layer.
- Produce evidence that a process is being followed.
- Do not contribute to productivity.
- This has not changed in decades.

Agenda

- Introduction
- Understanding Agile
- Role and Goal of "QA"
- Historical Role of QA
- QA in Business
- A Word About Development Processes
- QA in the Context of Development Processes
- Rethinking the Quality Abstraction
- An Implementation Example with Scrum
- Conclusion

A Word About Development Processes

- Relationship Between Management Methods and Development Methods
- De-Coupled Methods
- Software Methods

Relationship Between Management and Development Methods

- Hardware can be designed and manufactured in any one of several ways:
  - Design can be on paper, or using Computer Aided Design (CAD) systems.
  - Manufacturing can be by skilled artisans or can employ automated systems.
- These are the "development methodologies".
- Tools and tool control, inspection, inventory control, materials ordering, environmental controls, organizational needs.
- These are "management methodologies".

De-Coupled Methods

- The development and management methodologies, therefore, are distinct.
- Not completely de-coupled, however one does not dictate the other.
- They must:
  - complement and support one another.
  - work together to achieve business goals.
- Desirable to be optimized to work in the same business and operations strategy models.
- Fundamentally, whether blueprints are drawn by hand or by CAD is not dictated by how the flow of material is controlled through the plant.

Software Methods

- In the software world, for example, CMMI® doesn't care what development methodology is used.
- CMMI® doesn't dictate use of the "Waterfall" or Spiral models, or imply that XP is better than Scrum, "Crystal Light", and so on.
- Distinguishing the software *development* methodology from the software *management* methodology eliminates one of the barriers to managing QA in lightweight development environments.

Agenda

- Introduction
- Understanding Agile
- Role and Goal of "QA"
- Historical Role of QA
- QA in Business
- A Word About Development Processes
- QA in the Context of Development Processes
- Rethinking the Quality Abstraction
- An Implementation Example with Scrum
- Conclusion

QA in the Context of Development Processes

- Role of QA in the Development Process
- Responsibility of QA in the Development Process
- QA in Support of Development
- QA Distilled
- Processes within Lightweight Environments

Role of QA in the Development Process

- QA's role in the development process is fairly simple.
- Standards, methods and processes need to be followed and need to work well for the project.

Responsibility of QA in the Development Process

- QA is responsible for ensuring:
  - project's methodologies are taught to new developers on the project;
  - methods are followed by everyone, and
  - QA activities for projects are planned and not spontaneous.
- QA must:
  - measure the effectiveness of the methods,
  - provide visibility to management via appropriate metrics from prior project QA experience, and
  - know when the methods need to be adjusted.
- A person independent of the political and organizational chain-of-command are recommended to avoid conflicts-of-interest to achieve appropriate objectiveness from the product and its stakeholders.

QA in Support of Development

- QA must support the project in achieving the intended benefits of the standards the project sets for itself.

- If the project's processes and activities do not promote or support its standards, policies, or methods, it's QA's job to bring this disconnect to the attention of the people who can make appropriate changes.

QA Distilled

- QA boils down to making sure that the things that need to take place can happen and are happening, and that when they don't they get fixed.
- Everything else is technique.
- "Keeping things simple" is critical to a well-formed abstraction.
- When QA is distilled to the above statements, possibilities are created regarding how to look at the organization's QA processes so that they can operate in any environment.

Processes within Lightweight Environments

- Lightweight development doesn't mean there are no:
  - requirements management,
  - QC,
  - QA,
  - CM,
  - project planning,
  - project tracking, or
  - reviewing of designs and work.
- Development without those things would be called ***stupid*** programming, not ***agile*** programming.

Agenda

- Introduction
- Understanding Agile
- Role and Goal of "QA"
- Historical Role of QA
- QA in Business
- A Word About Development Processes
- QA in the Context of Development Processes
- Rethinking the Quality Abstraction
- An Implementation Example with Scrum
- Conclusion

Rethinking the Quality Abstraction

- Challenges of Current (typical) QA Approaches
- Necessary Value and Effectiveness of QA
- Real or Perceived "Pro-Active" Effort
- QA Processes Needed by the Market and Development
- Technology for Real-Time Analysis
- Transforming the Abstraction
- Desired "Target" Abstraction
- Section Summary

Challenges of Current (typical) QA Approaches

- Many QA processes rely on generating
  - artifacts,
  - evidence, and
  - other labor-intensive "bread crumbs"
  - tangential to the work being done on the product itself.
- These tangential efforts rely on the same people as development and therefore cannot occur on top of production, therefore increasing the amount of time it takes to carry out a project.
- **Stove-Piped** ☜ ☜ ☜

Necessary Value and Effectiveness of QA

- This approach to the QA process:
  - relegates QA to the role of policing and gate-keeping,
  - drastically minimizes the positive impact of the overall effectiveness of the QA program.
- One can seriously (and not without merit) question:
  - timeliness,
  - contribution, and
  - overall value
  - of QA
- … when the activities defined by or for QA purposes do not benefit the project.

Real or Perceived "Pro-Active" Effort

- "Proactive" QA is still seldom *pro*-active throughout the lifecycle of development.
- Proactive often means a level of effort before a project starts, followed by periodic or event-driven reactive activities that are only conducted as events unfold.
- It's this entire approach that needs "re-thinking".

QA Processes Needed by the Market and Development

- **NEED**: Process that:
  - ensure processes are matched to project objectives before the project gets under way
  - get into the detail of the standards and methods so that when the standards are followed they automatically generate the necessary "proof" of process compliance.
- **DON'T NEED**: Processes that
  - simply create automated markers and flags, or
  - reinvent the "wheel"
- **INSTEAD**:
  - approaches that enmesh metrics and data generation into the development process so that the successful output of the process is only possible if the process was properly followed.

Technology for Real-Time Analysis

- Use the technology of the development process, tools, and standards as the medium to collect process and tracking data

- Instead of policing the processes through post-mortem artifacts, QA could be free to analyze the effectiveness of processes in real time and make adjustments.

Transforming the Abstraction

- The actual abstraction transformation is simple.
- Instead of focusing the quality process on the effort of proving a formal process is followed, ensure that the processes are:
  - effective,
  - productive, and
  - valuable to the goals of the business, and
- Create production methods that produce the evidence as a by-product of the effort rather than a separate activity.

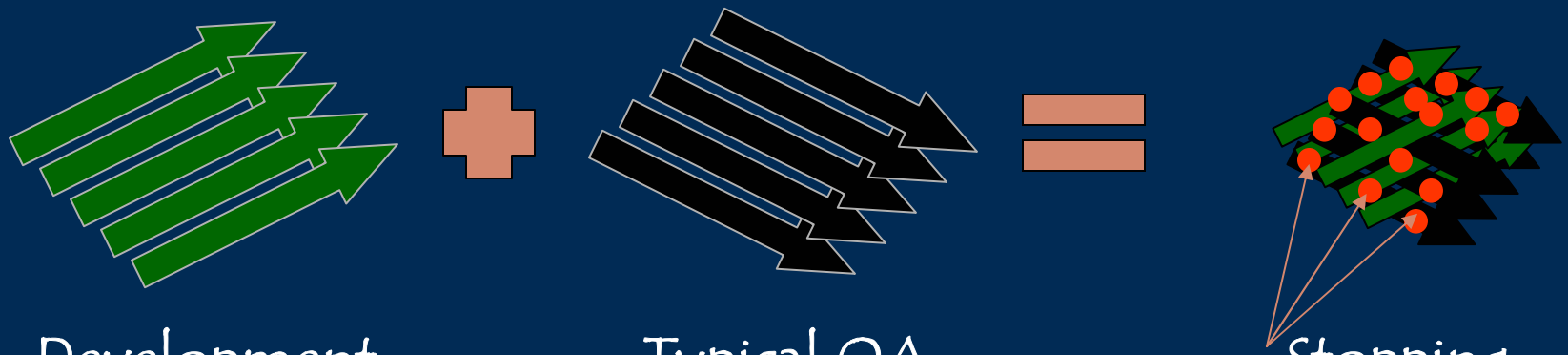
Desired "Target" Abstraction

- Transforming FROM:
  - A reactive, investigative, and stove-piped approach
- TO:
  - a productive, business-driven, value-focused umbrella of activities that improve the development effort
- Will achieve the "rethinking" of the QA abstraction that is necessary for lightweight development methods.

Section Summary

Go from this:

- Ordinary implementation of QA in development environments.



Development Processes

Typical QA Processes

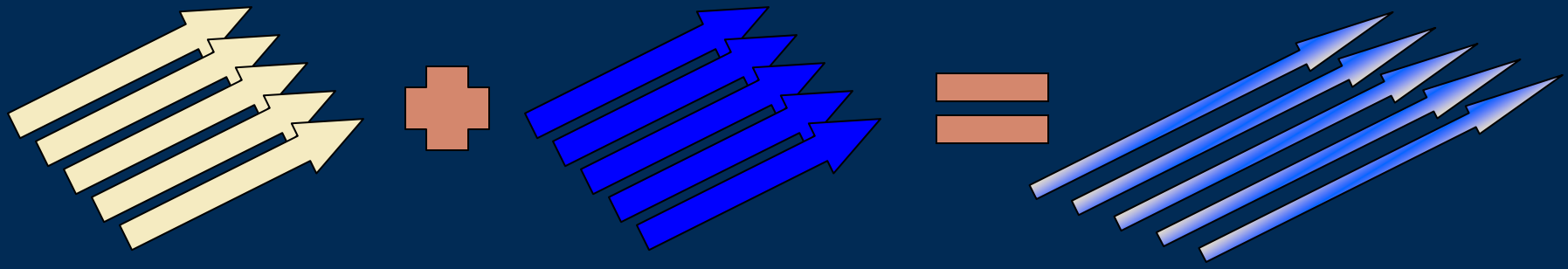
Stopping Points

- QA processes are in super-imposed onto development processes.
- Add a layer of effort not in-line with productivity.

Section Summary

To this:

- Preferred implementation of QA in development environments.



Development Processes

Agile QA Processes

Ø Stopping Points

- QA processes are integrated into and aligned with development, increasing development productivity.
- Contributes to capacity and value of company.

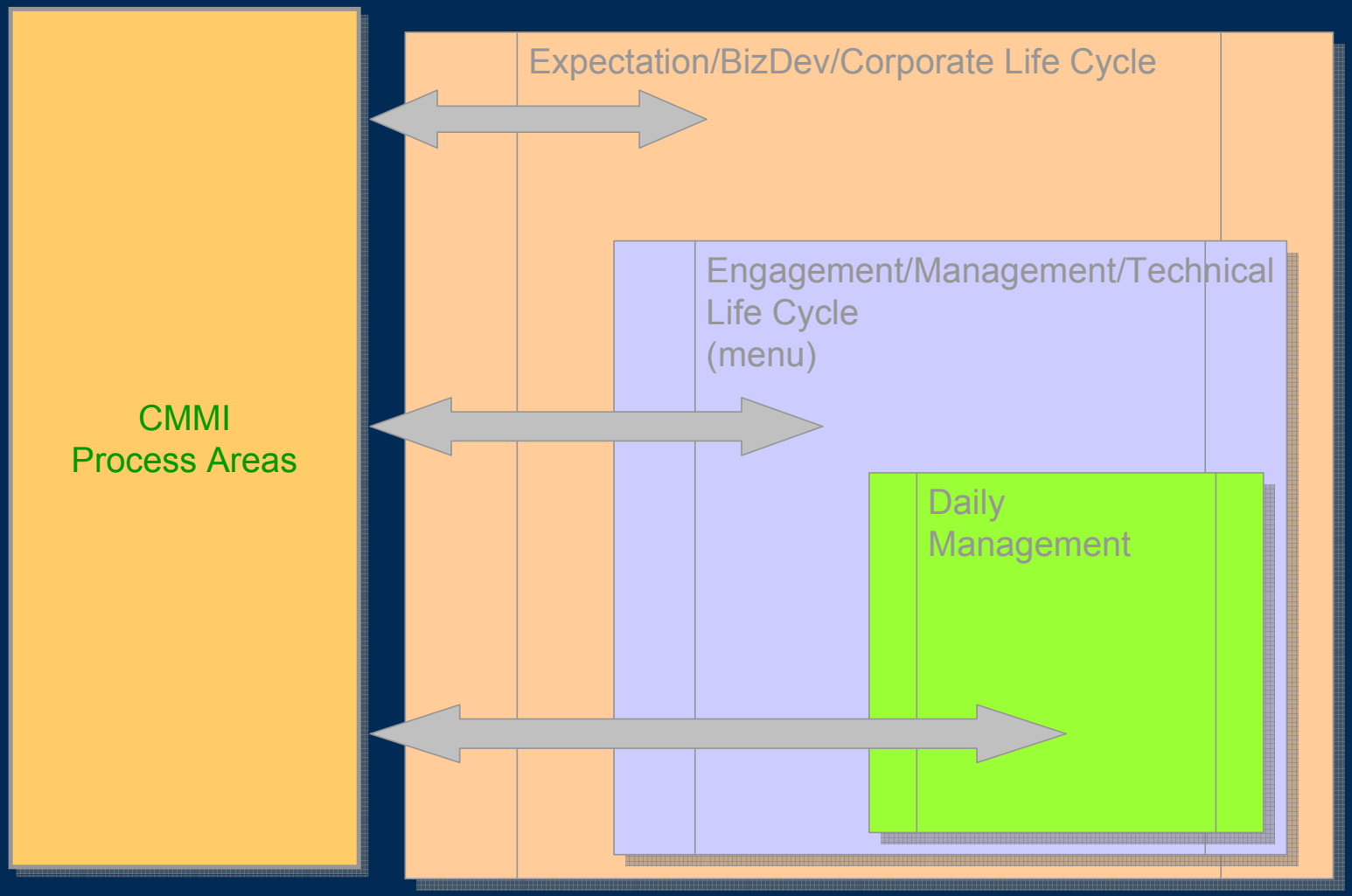
Agenda

- Introduction
- Understanding Agile
- Role and Goal of "QA"
- Historical Role of QA
- QA in Business
- A Word About Development Processes
- QA in the Context of Development Processes
- Rethinking the Quality Abstraction
- An Implementation Example with Scrum
- Conclusion

An Implementation Example

- Collect the desired practices that add the desired discipline.
- Find the actual work being done at a given location.
- Insert the practices into where the work is done.
- ID/Define life cycles in which actual work happens.
- Centralize redundant policies, processes, procedures and templates.

If *A* picture is worth…



…How much are *Several*?

Policy

Establishes that company projects will adhere to formal processes and states company's policy for quality values, quality work, and how these align with the company's mission and vision.

Quality Manual

Outlines what company does to ensure on-time, on-budget, fully featured/functional projects.

Expectation/ Corp /BizDev Fulfillment Life Cycle

Outlines the phases of every project @ company and scopes activities and deliverables within each phase. Establishes each project's parameters.

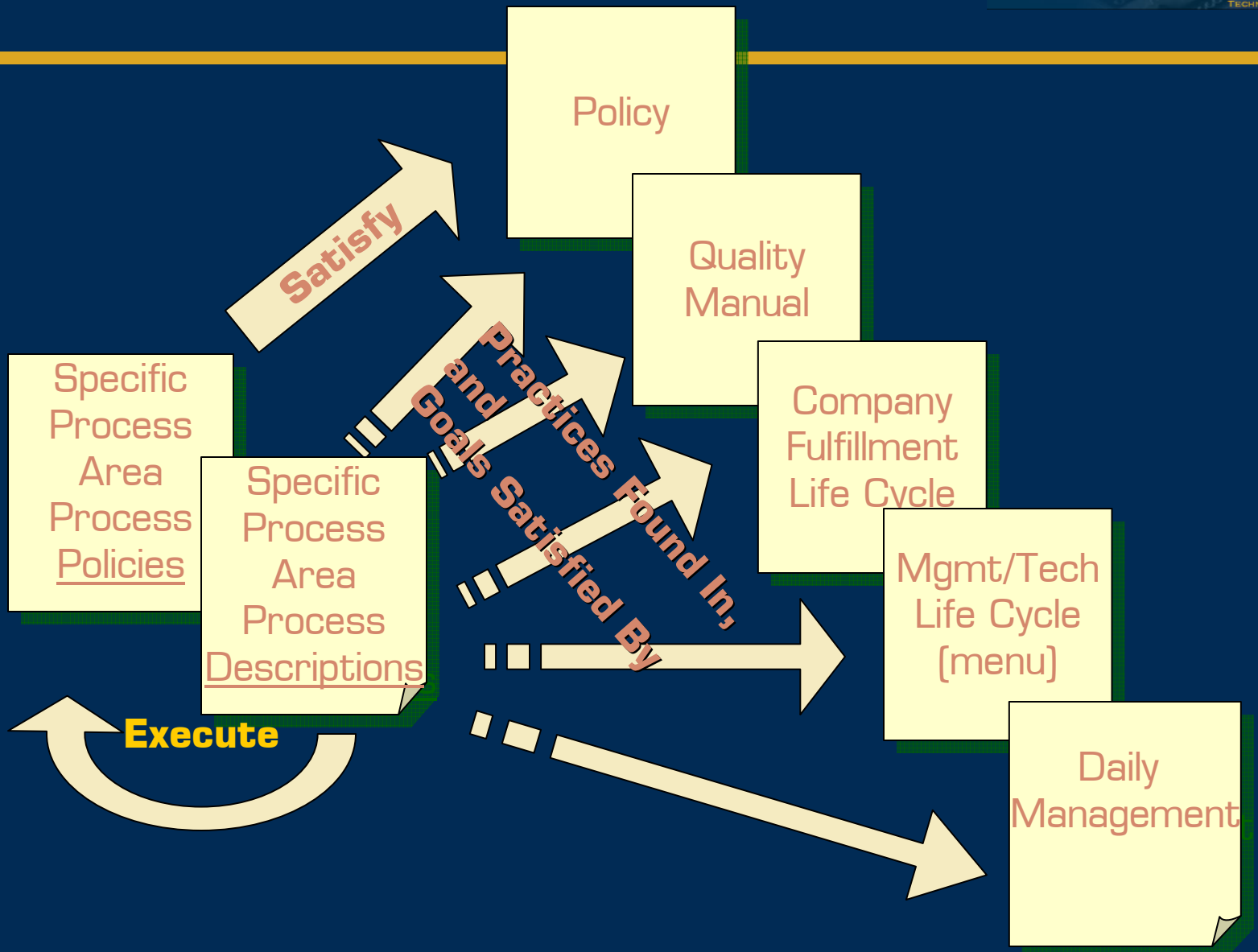
Engagement/ Mgmt/Tech Life Cycle (menu)

A menu of management or technical activities that each project can choose from as appropriate. Each project is required to identify a life cycle. This menu provides the list of what can be in a life cycle.

Daily/Weekly Management

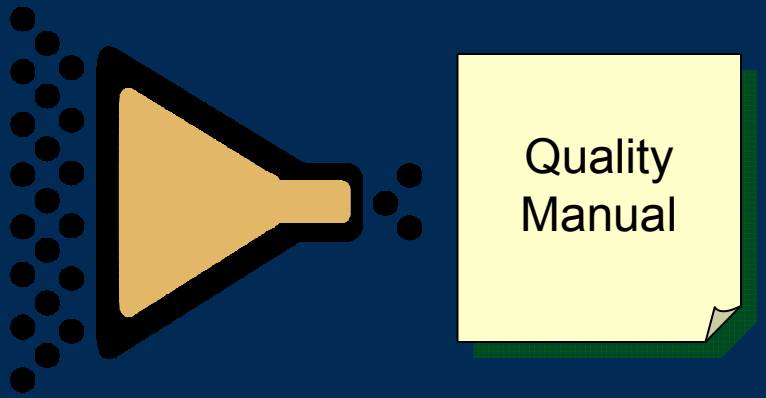
Specifies how projects are managed.

… AND THEN SOME…

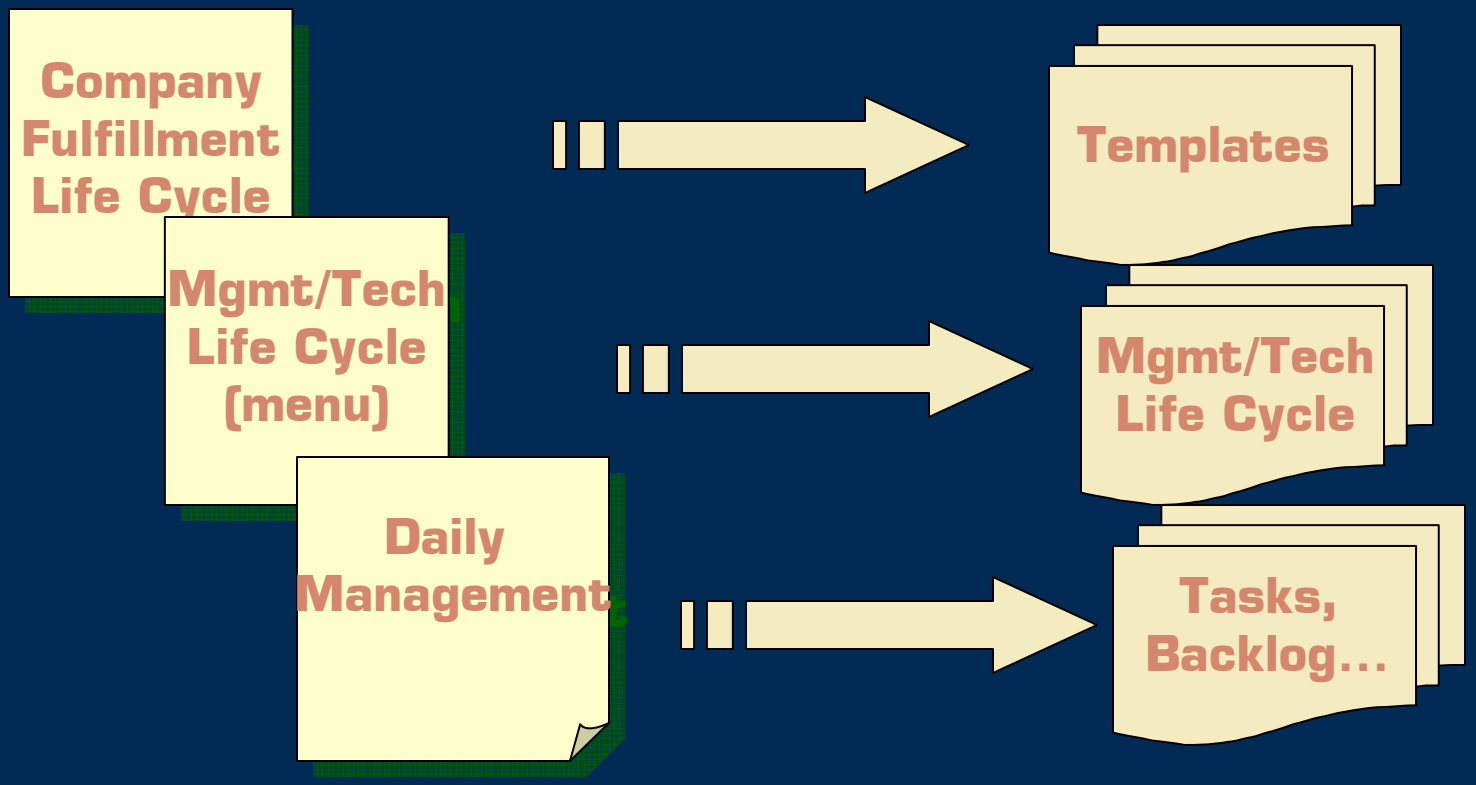


What's in the Quality Manual?

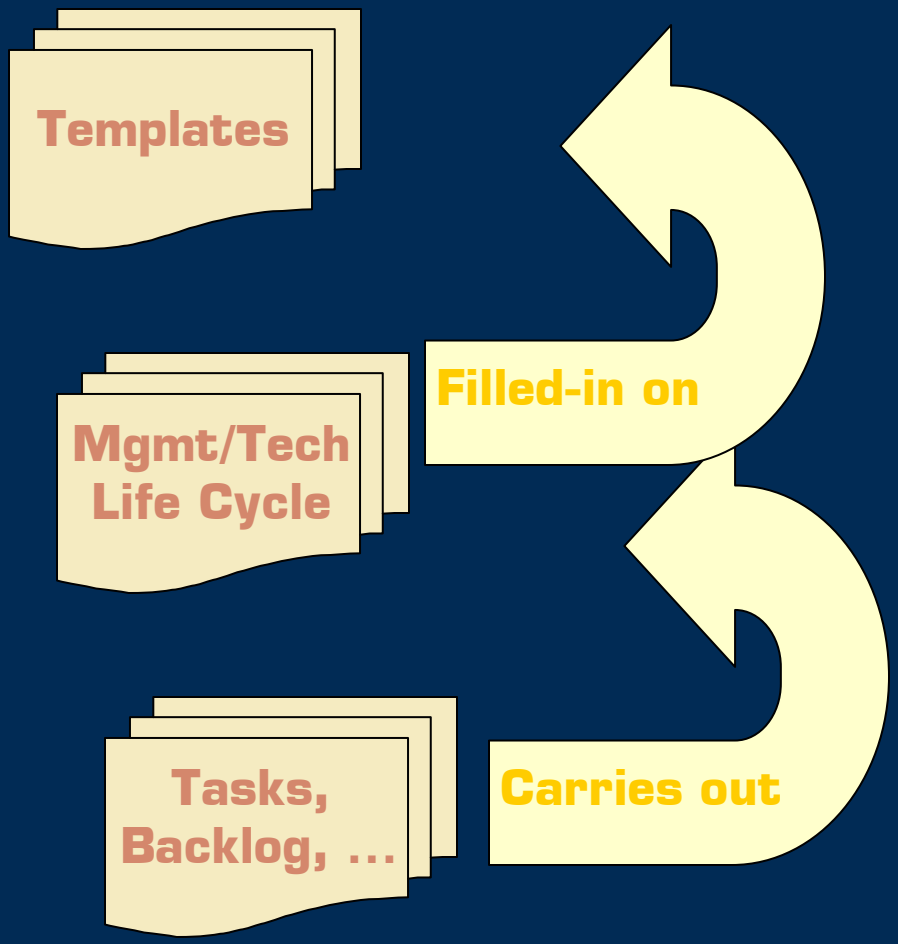
- **Explains how on each project, all company Processes:**
  - are planned-out and tailored from a single set of company processes
  - are assigned as someone's responsibility
  - are provided resources to be done
  - are assured of having people trained in them
  - have their work products configuration controlled
  - involve relevant stakeholders
  - are monitored & controlled
  - are objectively evaluated against applicable standards,
  - have performance reviewed with higher management, and
  - incorporate lessons learned for improvement



Work-Product Generation



Work-Product Interactions



company's Project Life Cycle

Fulfillment Life Cycle

- **Phase 1: Initial Analysis & Response**
- **Phase 2: Planning/Kick-Off**
- **Phase 3: Follow-Through**
- **Phase 4: Close-Out**

Phase 1 Concepts

Phase 1: Initial Analysis & Response

Phase 2: Planning/Kick-Off

Phase 3: Follow-Through

Phase 4: Close-Out

- Get from RFP to Award and/or from Award to Start
- Provides a business basis for going forward
- Provides requirements against which to manage the initial activities
- Scopes the project before details are known
- Breaks out of the Catch-22 of "when does the project start?"
- Allows for minimal mock-ups or prototyping/engineering analysis to obtain project requirements agreement.

Phase 2 Concepts

Phase 1: Initial Analysis & Response

Phase 2: Planning/Kick-Off

Phase 3: Follow-Through

Phase 4: Close-Out

- Identifies the project's:
  - Type
  - Management or Technical Life Cycle
  - Major Product and Document Deliverables
  - Major Tasks
  - Assignments, Roles and Stakeholders
  - Resources, Tools and Assets
  - Plans
  - Project Monitoring Events
  - Milestones
  - Required Training
  - Measures & Analyses

Phase 3 Concepts

Phase 1: Initial Analysis & Response

Phase 2: Planning/Kick-Off

Phase 3: Follow-Through

Phase 4: Close-Out

- All detailed engineering and provisioning of the solutions and products
- Execution of the entire Management or Technical Life Cycle
- From Design through Delivery and Installation
- Can be iterative with Phase 2
- All phases of the daily process through Closure

Phase 4 Concepts

Phase 1: Initial Analysis & Response

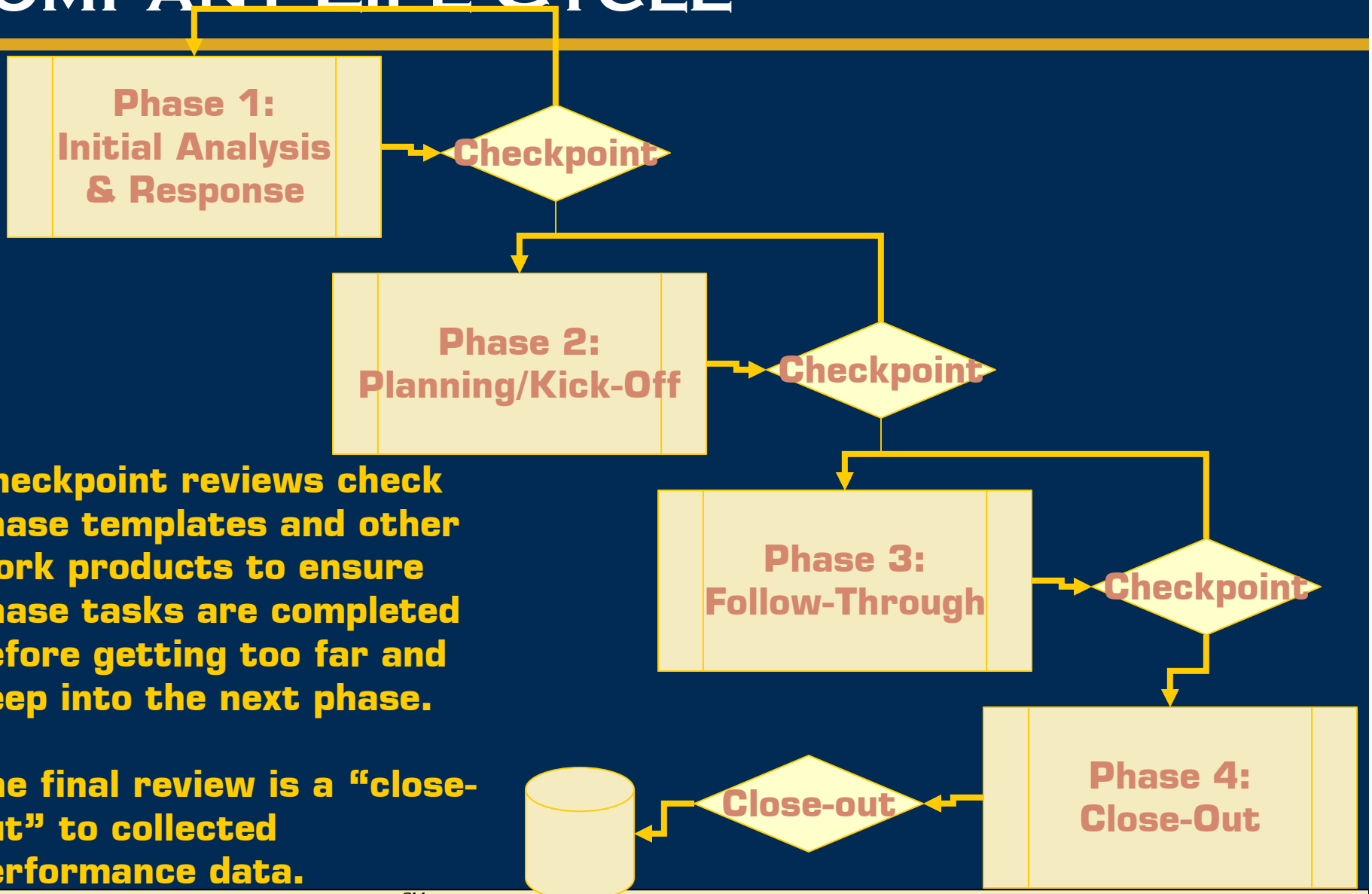
Phase 2: Planning/Kick-Off

Phase 3: Follow-Through

Phase 4: Close-Out

- Opportunity for Lessons Learned
- Final Administrative Checks
- Customer Feedback
- Final PPQA Checks & Audits
- Final CM Audits

PPQA Concepts in company Life Cycle

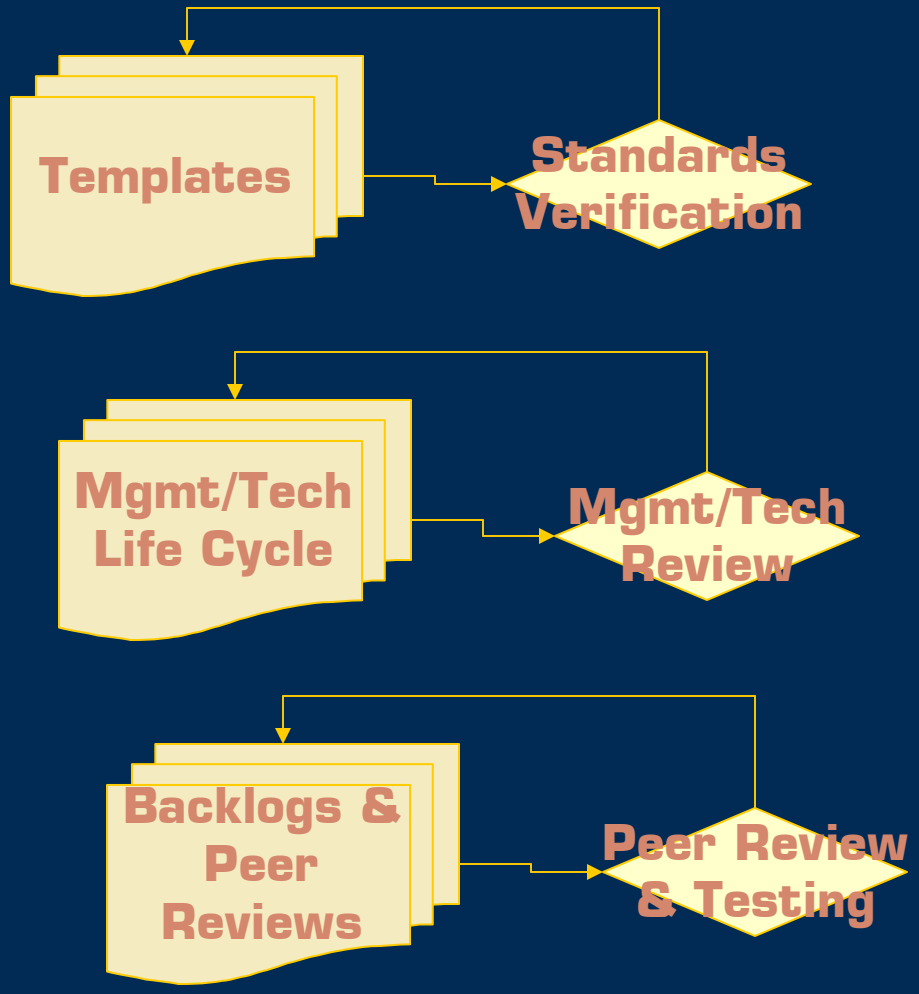


Checkpoint reviews check phase templates and other work products to ensure phase tasks are completed before getting too far and deep into the next phase.

The final review is a "close-out" to collected performance data.

PPQA Concepts in Work Products

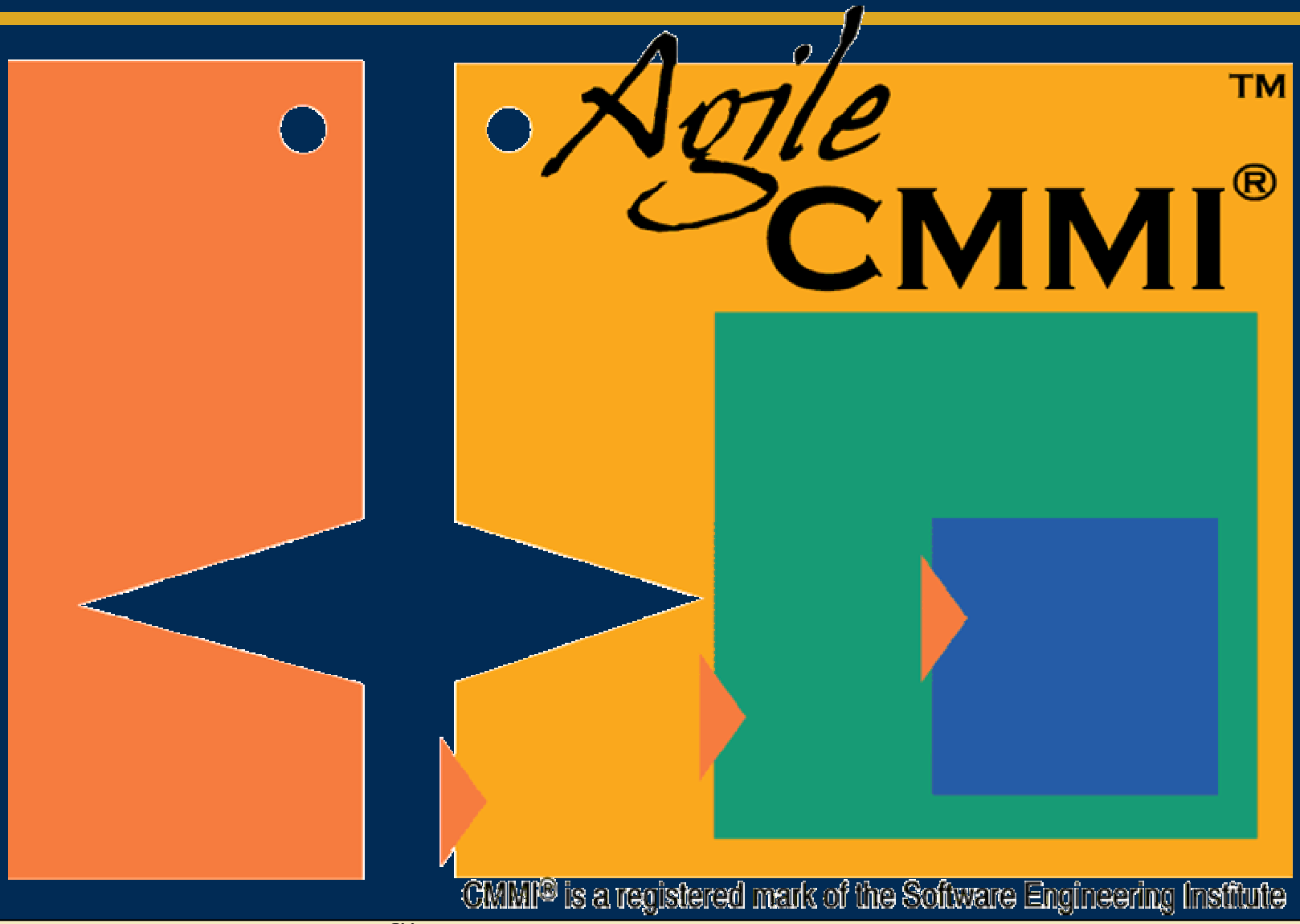
- Standards Verification performs process checks against company's own standards
- Engineering Reviews perform integrity checks on designs, analyses, and solutions
- Peer Reviews & Testing perform product checks on code and code-based work



All Other Processes

- All other practices within process areas have been distributed into and made seamless with company planning and engineering activities.
- Some practices are performed once and passed through with each project review.
- Some practices are addressed by merely including an item on a meeting agenda.

Agile CMMI



QA's Job in the New Abstraction

- QA's real job is to ensure certain information is generated to assure that processes are followed.
- There may be no need for QA to get mired in the specifics of development.
- By collaborating with developers on producing what QA needs,
  - the every-day hour to hour activities of development can become part of development activities and
  - QA can be left to monitor the overall effectiveness of the project's processes and feed back process improvements.

Scalable QA in a Scalable Project

- In a well integrated project:
  - generating the data QA needs would merely be a report that runs every so often querying certain tables and build repositories.

- A QA program at this level of abstraction is:
  - infinitely scalable to any project
  - as long as there's the will to cooperate
  - for the purposes of benefiting the business.

CMMI with Scrum

- Product Backlog and Planning
- Sprint Backlog and Planning
- Resource Allocation
- WBS
- Daily Team Meetings
- Peer Reviews and Inspection
- Sprint Review

Product Backlog and Planning

- The product backlog is defined by the **product owner** and managed by the Scrum master.
- Defines High Level Requirements and sets priorities.
- Defines high level work break down structure.
- May define high level release schedule.

- REQM
- PP
- PMC
- CM
- GP 2.2, 2.3, 2.4, 2.7
- (RD, TS, PI, IPM, RISK, DAR)

Sprint Backlog and Planning

- Breaks the product goals down into demonstrable goals. This is usually at the use case level.
- Tasks are broken down into hour-based estimates, anything over 16 hours was broken down into smaller pieces.
- The team creates tasks, estimates and determines who is going to do what, everyone commits to the feasibility of the plan.
  - What can be done in 30 days with the resources we have at our disposal?

- REQM
- PP
- PMC
- CM
- GP 2.2, 2.3, 2.4, 2.6, 2.7
- (RD, TS, PI, VAL, VER, IPM, RISK, DAR)

Resource Allocation

- Managed by the team, as members commit to getting the work done.
- Members **can play many roles at the same time:**
  - Developer, Architect and DBA
  - Developer, Tester and Requirements Analyst
- Member are committed to the project and external noise is minimized.
- The Scrum Master helps alleviate resource contention and noise.

- REQM
- PP
- PMC
- MA
- CM
- GP 2.2, 2.3, 2.4, 2.7
- (RD, TS, PI, IPM, RISK, DAR)

Work Breakdown Structure

- A Product Goal can be broken down into many **use cases**
  - "The application needs to contain a shopping cart"
- A Sprint Goal satisfies a use case
  - "Allow a registered use to put items into their shopping cart"
  - "Allow a user to update the quantities in the shopping cart"
- Each sprint goal is demonstrable, releasable functionality.
  - Show that this use case works, and has been tested and could be released as functionality

- REQM
- PP
- PMC
- CM
- GP 2.2, 2.3, 2.4, 2.7
- (RD, TS, PI, IPM, RISK, DAR)

Daily Team Meetings

- Quick 15-30 Minute Stand up Meetings.
- Answer 3 Questions:
  - What have you done since the last meeting ?
  - What are you going to do before our next meeting ?
  - What issues are you having that are impeding progress ?
- Daily Inspection and Visibility into team progress.
- Daily Issues Management and Resolution.
- Daily Project Command and Control within the self managing team.

- REQM
- PP
- PMC
- MA
- PPQA
- CM
- GP 2.2, 2.3, 2.4, 2.6, 2.7, 2.8, 2.9, 2.10
- (RD, TS, PI, IPM, RISK, DAR)

Peer Review and Inspection

- Peer reviews keeps the team members honest.
- Peer reviews are about **mentoring, not policing**.
- Complete **checkpoints and tollgates** along the project road map that can be done iteratively and kept **non-invasive.**

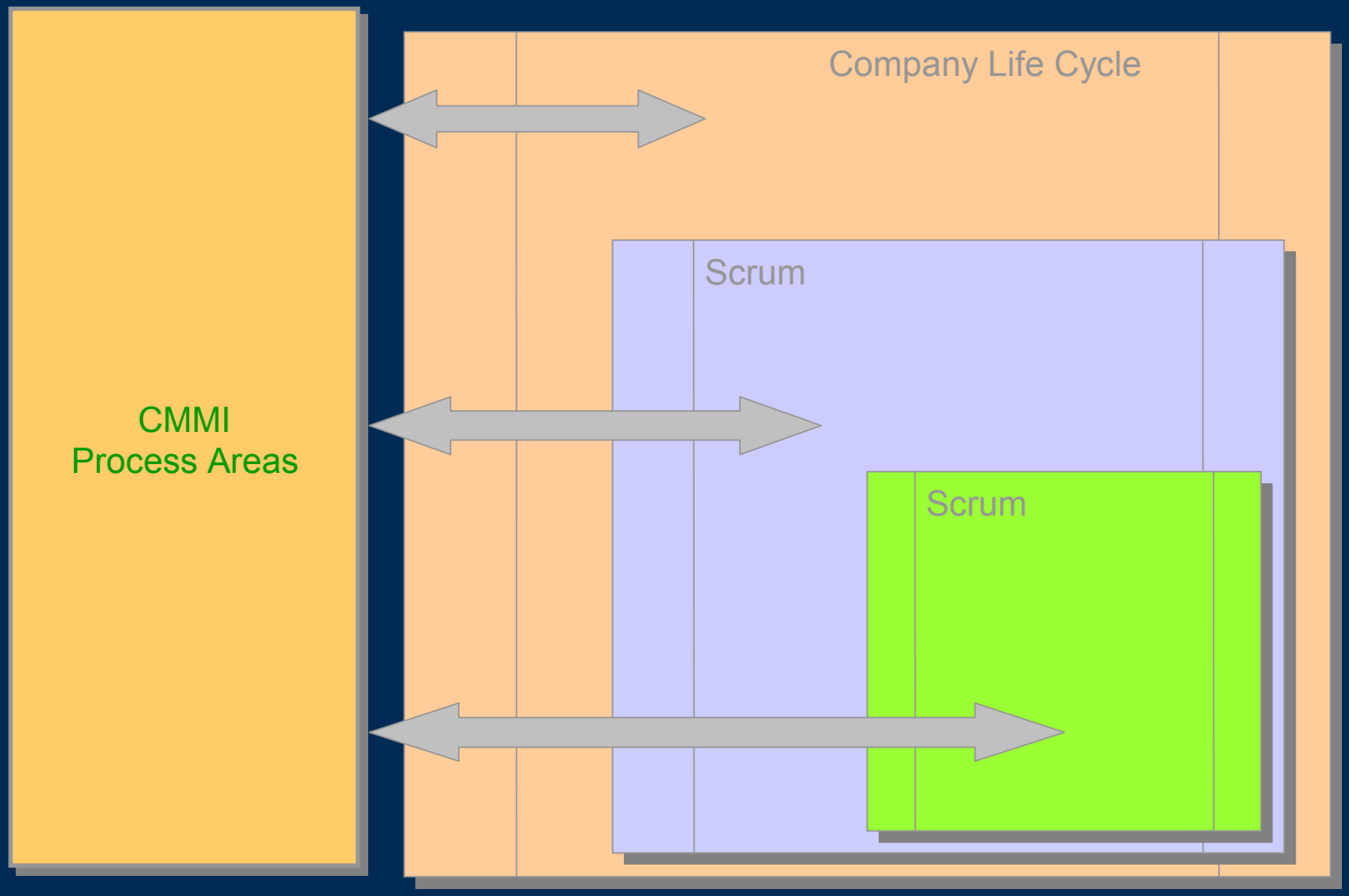
- REQM
- PMC
- MA
- PPQA
- CM
- GP 2.6, 2.7, 2.9, 2.10, 3.2
- (RD, TS, PI, VAL, VER, IPM, RISK, DAR)

Sprint Review

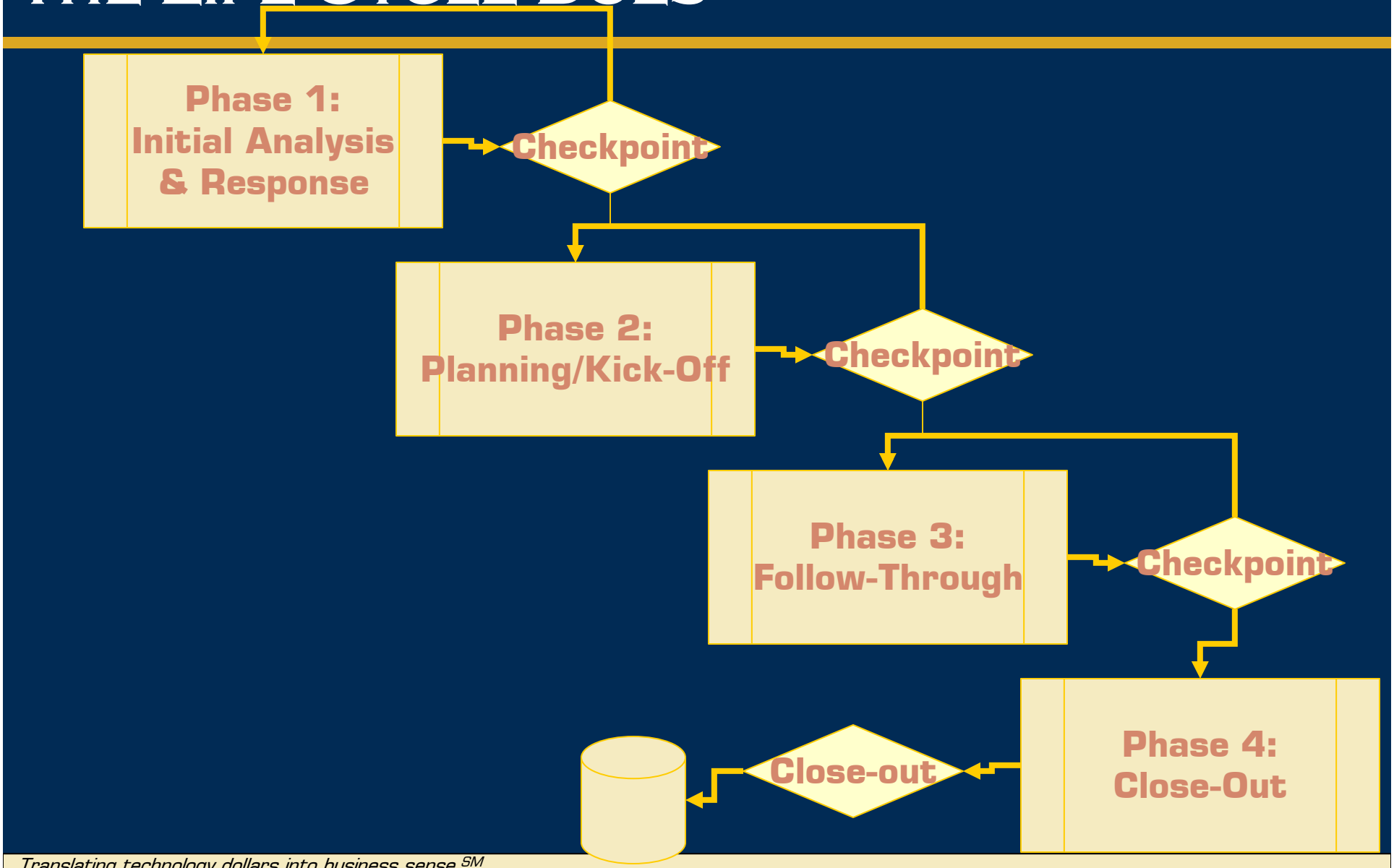
- The Sprint review is a form of validity check-it is determined that the right product is being built.
- Covers whether the product was built right because a working version of the product is giving a viewing to the product owner.
- Product Owner (s) decides if functionally and quality are sufficient to be released

- REQM
- PMC
- MA
- PPQA
- CM
- GP 2.6, 2.7, 2.9, 2.10, 3.2
- (RD, TS, PI, VAL, VER, IPM, RISK, DAR)

Scrum Summary



What Scrum Doen't Cover, The Life Cycle Does



Agenda

- Introduction
- Understanding Agile
- Role and Goal of "QA"
- Historical Role of QA
- QA in Business
- A Word About Development Processes
- QA in the Context of Development Processes
- Rethinking the Quality Abstraction
- An Implementation Example with Scrum
- Conclusion

Conclusion

- Reversing Common Misconceptions About Agile
- Re-Cap
- References
- Q&A

Reversing Common Misconceptions About Agile

- Lightweight/Agile is *not* about eliminating processes.
- Lightweight/Agile *is* about sufficient processes to achieve the project's objectives, but only those processes that are necessary.
- QA and other development processes have not kept up with the changes in development methods or technologies.
- Agile processes have everything the agile project needs to get accomplished.

Re-cap

- History of QA Abstractions
- Context of the Role of QA
- Mutual Distaste Between Lightweight Developers and Process Discipline Practitioners
- Create processes that Promote Productivity
- Middle Ground Where Diligently Followed Lightweight Methods Find Mutually Beneficial Ground with Process Discipline
- Result: Excellent Software of Very Good Value.

Re-cap

- Today's s/w market is not that of 15+ years ago.
- Traditional QA fails on s/w because the manufacturing model for development is a failure.
- QA must be integrally aligned with development processes and business goals to survive in today's market.
- Agile development processes will survive because they're more consistent w/SW realities.
- Agile can be disciplined if discipline can be agile.

References

- Beck, Kent. *Extreme Programming Explained: Embrace Change*, Addison-Wesley, 2001.
- Glazer, Hillel. "Dispelling the Process Myth," *CrossTalk*, November 2001, Vol. 14 No.11, pp. 27-30.
- http://alistair.cockburn.us
- http://c2.com/cgi/wiki?XpAndTheCmm
- http://c2.com/cgi/wiki?CategoryPattern
- http://c2.com/cgi/wiki?QaIsNotQc
- http://www.extremeprogramming.org
- http://www.martinfowler.com/articles/newMethodology.html
- http://www.sei.cmu.edu

Q & A

Hillel Glazer, Principal & CEO
Entinex, Inc.
+01.301.384.4203
spi @ entinex.com
www.entinex.com

Chris Hagenbuch, PMP, CPSM
Dir. Solutions Engineering, InfoZen
301.980.0602
chagenbuch @ infozen.com