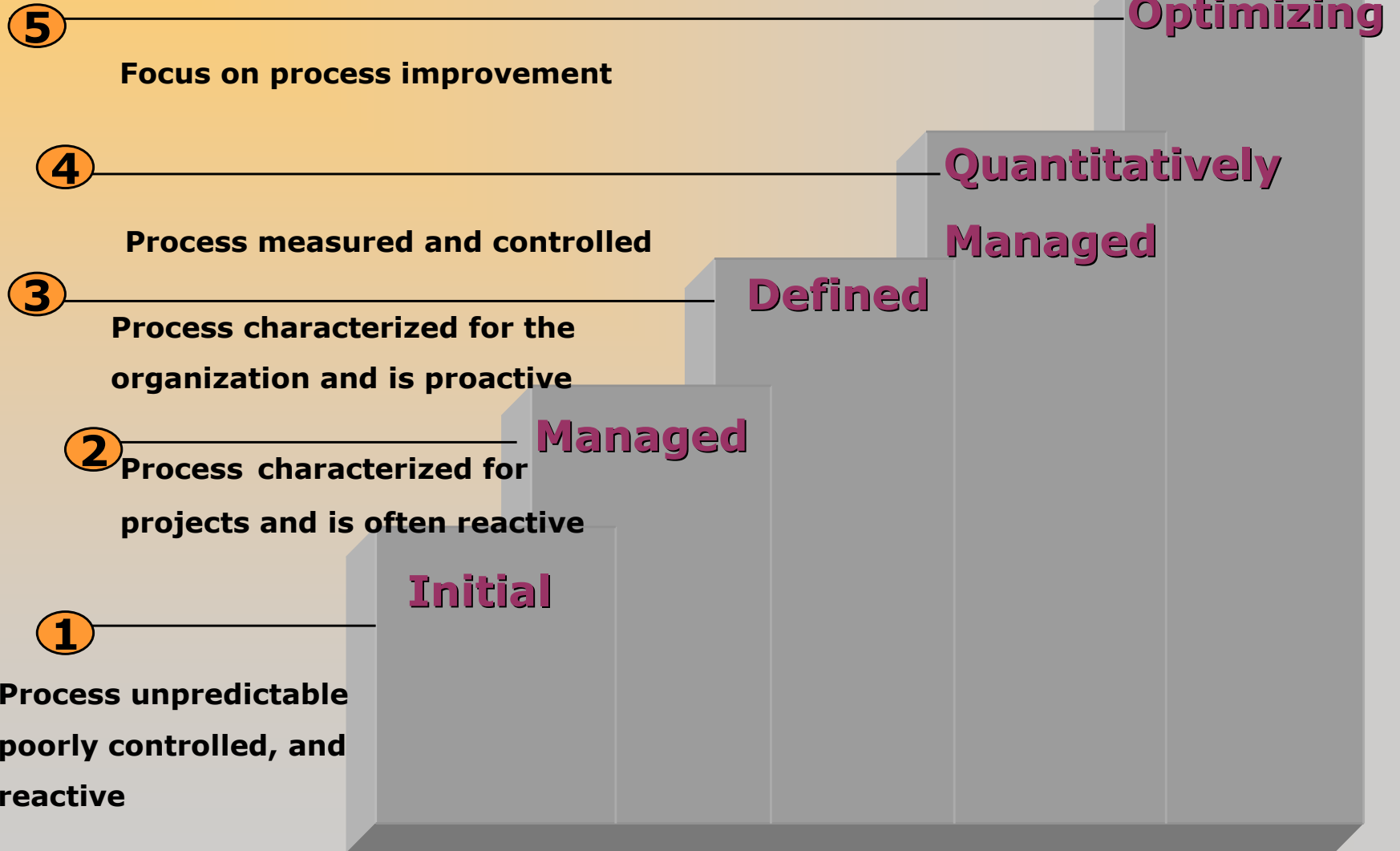


CMMI - The AGILE Way

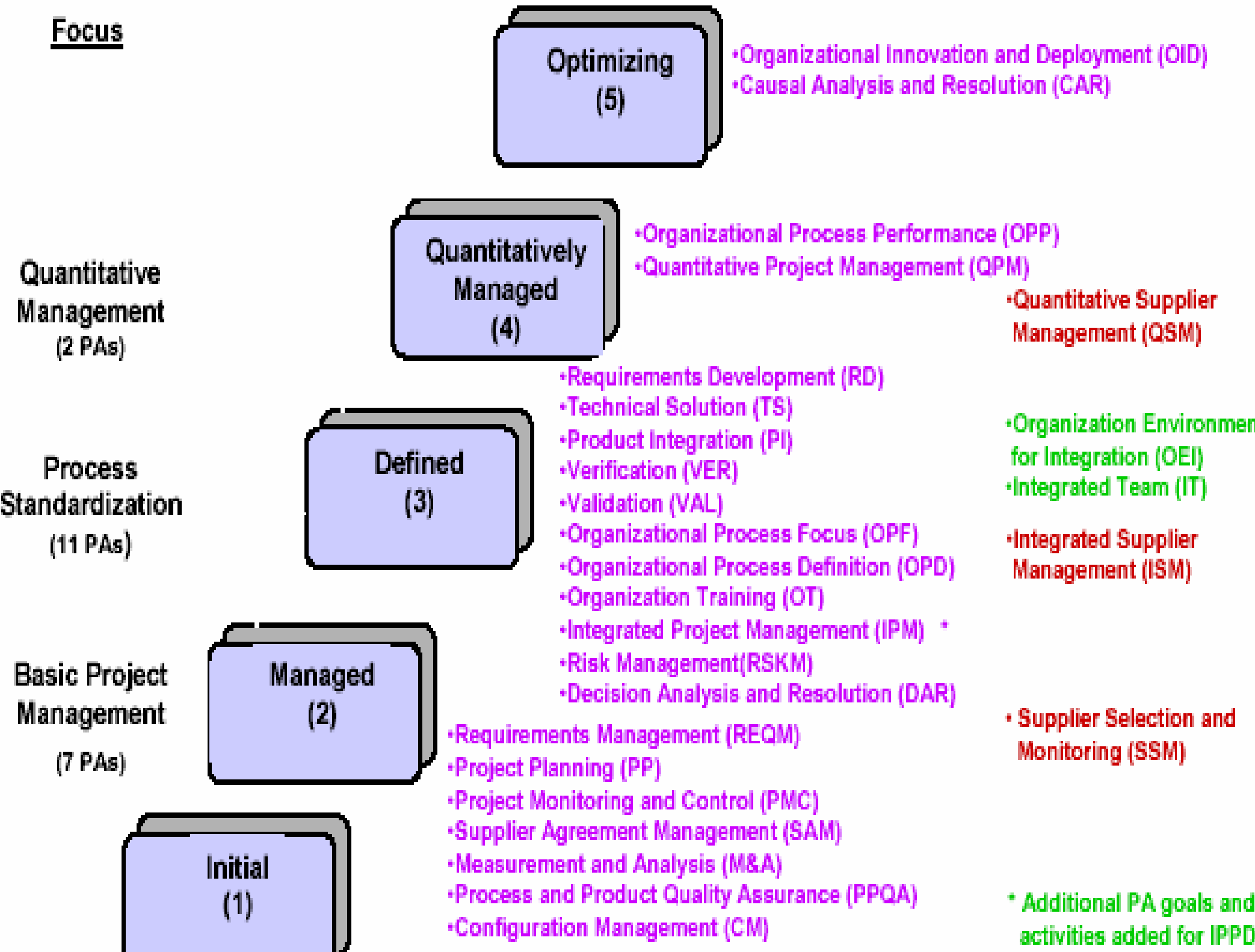
By

Hitesh Sanghavi

The Maturity Levels



Focus





PA Categories

CMMI[©]

Process Management

- Organizational Process Focus
- Organizational Process Definition
- Organizational Training
- Organizational Process Performance
- Organizational Innovation and Deployment

Project Management

- Project Planning
- Project Monitoring and Control
- Supplier Agreement Mgmt.
- Integrated Project Mgmt.
- Risk Management
- Quantitative Project Mgmt.

Engineering

- Requirements Management
- Requirements Development
- Technical Solution
- Product Integration
- Verification
- Validation

Support

- Configuration Mgmt.
- Process and Product Quality Assurance
- Measurement & Analysis
- Decision Analysis and Resolution
- Causal Analysis and Resolution

IPPD

- Organizational Environment for Integration
- Integrated Teaming

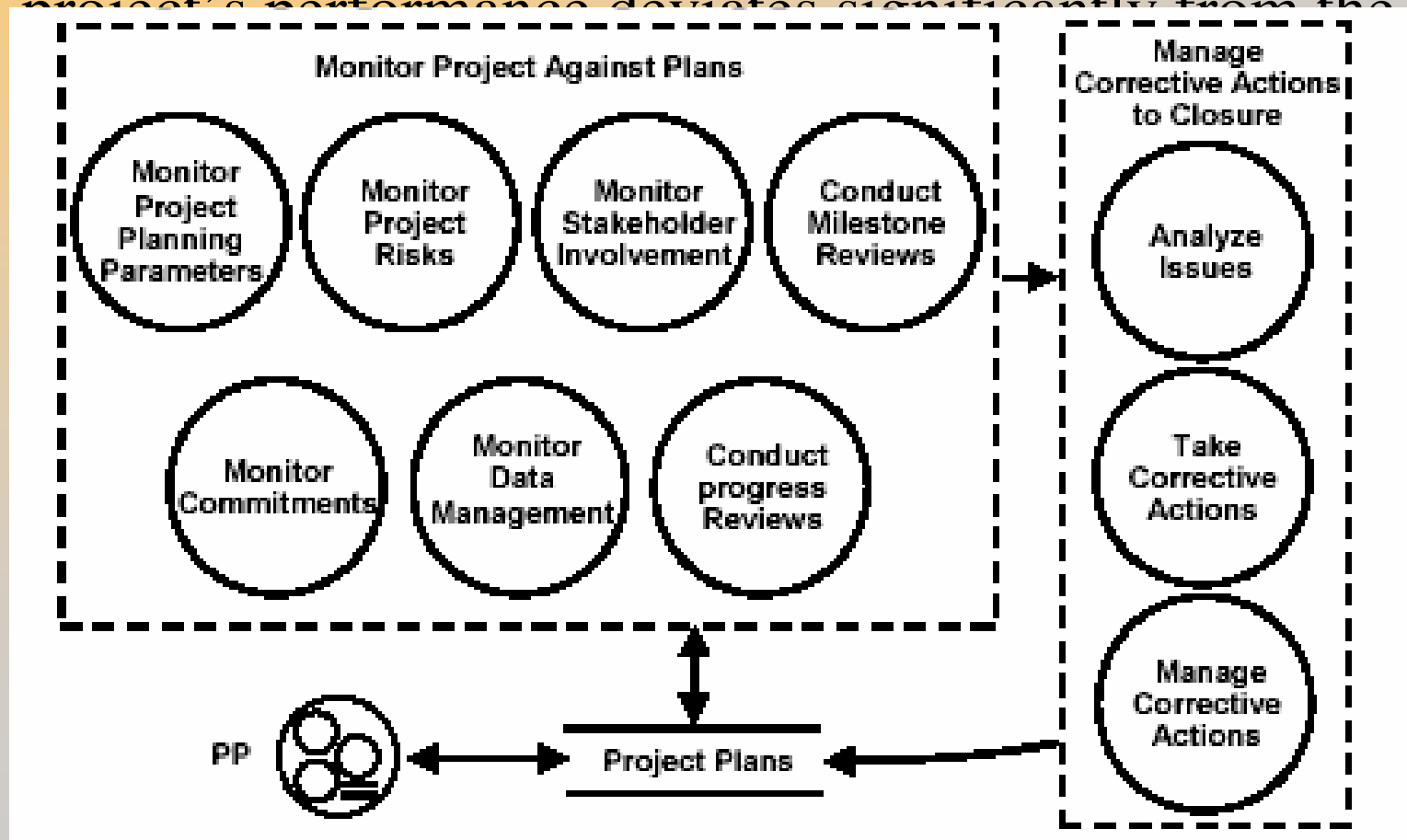
Supplier Sourcing

- Supplier Selection and Monitoring
- Integrated Supplier Management
- Quantitative Supplier Management



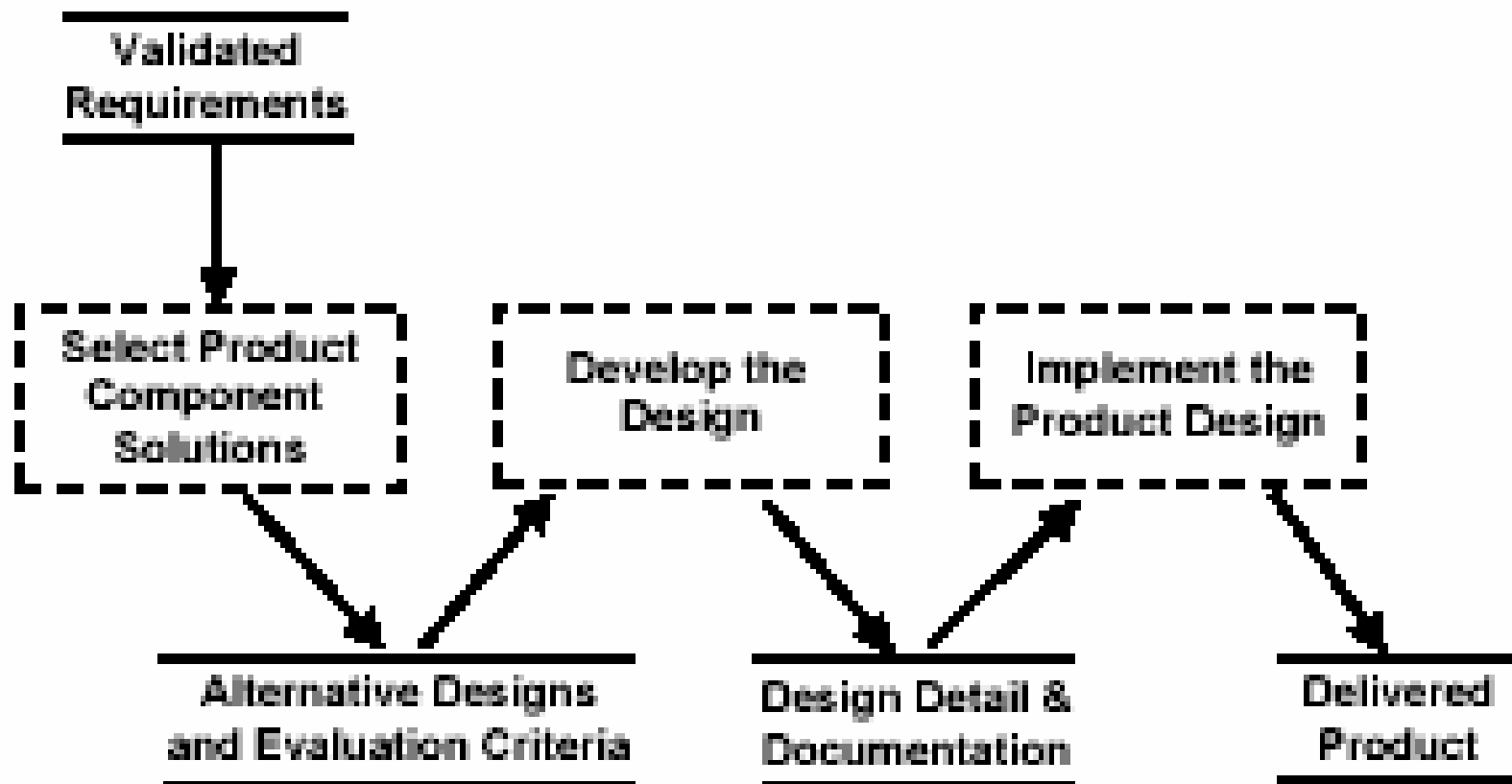
Project Monitoring and Control

- ★ Purpose: Provide understanding into the project's progress so that appropriate corrective actions can be taken when the project's performance deviates significantly from the plan.





Technical Solution – Context 1





CMMI - Generic Practices



Generic practices for all maturity level 2 process areas:

GP 2.1: Establish an Organizational Policy

GP 2.2: Plan the Process

GP 2.3: Provide Resources

GP 2.4: Assign Responsibility

GP 2.5 : Train People

GP 2.6: Managed Configurations

GP 2.7: Identify and Involve Relevant Stakeholders

GP 2.8 : Monitor and Control the Process

GP 2.9: Objectively Evaluate Adherence

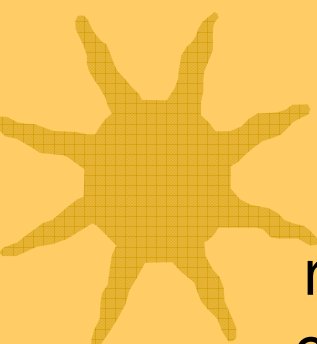
GP 2.10: Review Status with Higher Level Management

GP 3.1: Establish a Defined Process

GP 3.2: Collect Improvement Information



A Brief Overview of Agile+



Agile methods is a unique and effective hybrid approach (Extreme Prog-XP, SCRUM, Crystal,Lean Development) to SW development that retains the core elements of (XP). IT recommend very down-to-earth, actionable practices such as continuous integration, test-first programming, and refactoring



Agile development is:

- Essential streamlined Highly disciplined processes
- No nonsense, intensely practical
- Bias towards action
- Flexible, adaptable and speed-to-market
- Execution with swiftness and precision



AGILE Advantages



AGILE provides a uniquely effective approaches for...

- ★ optimizing development team productivity (OID)
- ★ minimizing risk (RSKM)
- ★ ensuring end-user satisfaction and support (RM)
- ★ rapidly and continuously delivering high-value working software to end-users (IPM)
- ★ delivering the ROI that business and technical leaders alike demand from software projects(MA)
- ★ The ability to deal with incomplete and unstable requirements
- ★ More effective control over schedule and costs
- ★ The right balance between agility and rigor
- ★ A much more enriching and satisfying way to develop software
- ★ A happier customer

Extreme Programming

(Kent Beck, Ron Jeffries, and Ward Cunningham)



- ★ XP's target is small to medium sized teams building software with vague or rapidly changing requirements. XP teams are typically collocated and have fewer than 10 members.
- ★ XP's critical underlying assumption is that developers can obviate the traditional high cost of change using technologies such as objects, patterns, and relational databases, resulting in a highly dynamic XP process.
- ★ XP teams typically deal with requirements changes through an iterative life cycle with short cycles.



The XP life cycle



The XP life cycle has four basic activities:

- ★ Coding (TS), testing(V-V), listening(RM-PMC,RSKM), and designing(TS).

XP dynamism is demonstrated through four values:

- ★ Continual communication with the customer and within the team; (GP 2.7,2.8,IPM)
- ★ Simplicity, achieved by a constant focus on minimalist solutions;(RD, TS SP2.4)
- ★ Rapid feedback through mechanisms such as unit and functional testing;(VV)
- ★ The courage to deal with problems proactively.(RSKM)



12 XP Elements -1 :

- ★ **1. Planning game:** Quickly determine the next release's scope(PP), combining business priorities and technical estimates(PP). The customer decides scope, priority, and dates from a business perspective, whereas technical people estimate and track progress(PMC).
- ★ **2. Small releases:** Put a simple system into production quickly. Release new versions on a very short (2-3-week) cycle. (OPD - SP1.2 [LC] and 1.3 [TG])
- ★ **3. Metaphor:** Guide all development with a simple, shared story of how the overall system works. (RM & RD, IPM, GP2.7)
- ★ **4. Simple design:** Design as simply as possible at any given moment.(RD,TS)



XP Elements -2 :



- ★ **5. Testing:** Developers continually write flawless running unit tests (V-V); customers write tests to demonstrate that functions are finished. “Test, then code” means that a failed test case is an entry criterion for writing code.
- ★ **6. Refactoring:** Restructure the system without changing its behavior to remove duplication(RM-SP1.4[RTM]), improve communication (GP 2.7), simplify or add flexibility.(RM-SP 1.5- [Id Inconsistencies])
- ★ **7. Pair programming:** All production code is written by two programmers (GP 2.3,2.4,2.5,2.7) at one machine. For a jelled team(IT), the effort increase(PMC) can be as little as 15% (MAA, GP2.2,2.8), while cycle time is reduced by 40 to 50 %.
- ★ **8. Collective ownership:** Anyone can improve any system code anywhere at any time.(GP 2.7,GP 2.6)



XP Elements -3 :



- ★ **9. Continuous integration:** Integrate and build the system many times a day (every time a task is finished). Continual regression testing prevents functionality regressions when requirements change. (PI)
- ★ **10. 40-hour weeks:** Work no more than 40 hours per week whenever possible; never work overtime two weeks in a row. (PP,PMC)
- ★ **11. On-site customer:** Have an actual user on the team full-time to answer questions. (IPM loose mapping)
- ★ **12. Coding standards:** Have rules that emphasize communication throughout the code.(TS)



XP Metrics



- ★ The basic XP management tool is the metric, and the metric's medium is the “big visible chart.” In the XP style, three or four measures are typically all a team can stand at one time, and those should be actively used and visible.
- ★ One recommended XP metric is “project velocity”—number of stories of a given size that developers can implement per iteration. (MA)



Can CMMI and Agile Co-exist



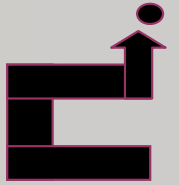
- ★ Success in any endeavor requires both agility and discipline
- ★ Two approaches to software development
 - Plan-driven (SW-CMMI, document-based, strong process)
 - Agile (XP, tacit knowledge, light process)
- ★ Agile and plan-driven proponents are believers
- ★ Both have strengths and weaknesses – balance is needed



AGILE - People over Process



- ★ Development is “of the people, by the people, for the people”
- ★ Separation of concerns is increasingly harmful



Philosophy of CMMI with Agile

The most important thing to understand about the CMMI is that it is all about achieving the goals.

The CMMI prescribes what must be achieved to have an effective process, not how to go about reaching the goals

Agile+ is unique in that it permits an organization to deploy agile methods with their attendant advantages, and simultaneously, achieve compliance with the CMMI

Illustration: See Goalwise Compliance



Balanced Methods are Emerging



★ Agile methods

- Crystal Orange
- DSDM
- FDD

★ Plan-Driven methods

- Rational Unified Process
- CMMI

★ Hybrid

- Boehm-Turner Risk-based
- Manzo (AgileTek) Code



AGILE Recommendations



- ★ Make incremental change to the requirements, project plan system, and the resulting artefacts to enable agility.
- ★ Strive for rapid feedback to ensure the project meets the needs of all the participants and stakeholder.
- ★ Manage with a purpose, performing only those tasks that add value to business processes supported by the system.
- ★ Travel light, discarding processes and artefacts that don't add enduring value to the product — a working software system.

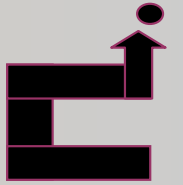


High Maturity with CMMI



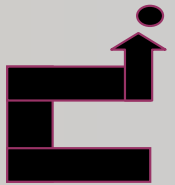
- ★ Innovation at level 5 argues for agile approaches
- ★ Having both agile and plan-driven standard processes allows marketplace agility
- ★ Application of Lean and Six Sigma techniques at high maturity levels eliminates non-value added processes and results in more agile tailor-up rather than tailor-down approaches

Mapping Agile+ to the CMMI

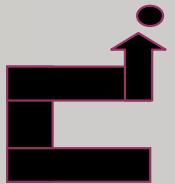


This map illustrates which Agile+ practices fulfill the goals of which Pas

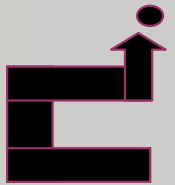
For the purposes of the map, the following abbreviations are used for the Agile+ practices



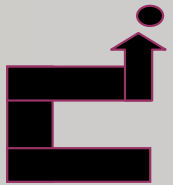
Agile+ Practice	Abbreviation
Customer at the Center of the Project	CUS
Flexibility to Meet Client's Special Needs	FLX
Business Process Analysis	BPA
Risk-Based Situation Audits	RSK
User Stories	STO
Story Actors	ACT
Iterative Development	ITR
Collective Ownership	OWN
Wall Gantts	GNT
Continuous Integration	CON
Componentized Architecture	ARC
Delphi-STE Estimation	EST
Design and Control of Interfaces	INT
Configuration Management	CMG
Relentless Testing	TST
Automated Contract and Regression Testing	CRT
Coding Standards	COD
Automatic Document Generation	ADG
Refactoring	REF
Pair Programming	PPG



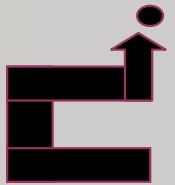
Process Area	Specific Goals	Agile+ Practice	Comments
Requirements Management	1. Manage Requirements	CUS FLX BPA STO ACT	CUS - Tight coupling with the customer ensures integrity of requirements. FLX - The flexibility of Agile+ allows for keeping the project in synch with requirements as they evolve. BPA - Business process analysis ties requirements to business need. STO - User stories are focused on being understandable to the customer. ACT - The use of story actors keeps focus on the characteristics of those who will actually use the software.
Project Planning	1. Establish Estimates 2. Develop a Project Plan 3. Obtain Commitment to the Plan	ITR OWN GNT EST	ITR - At each iteration planning session, detailed plans are created for the upcoming iteration. The overall, high-level project plan is also updated base on the experience of the previous iteration. OWN - In Agile+, the PM is a facilitator. The entire team is responsible for planning the iteration and optimizing the user of resources. GNT - Use of the Wall Gantt and its accompanying daily standup meeting creates an atmosphere where the project plan is evaluated and updated daily. EST - For mid-size project, formal Delphi estimation is used. For smaller projects, an informal group session may be sufficient for estimation. For larger projects, STE is added to the Delphi estimation.
Project Monitoring and Control	1. Monitor Project Against Plan 2. Manage Corrective Action to Closure	FLX OWN GNT	FLX - Both the adaptability of the Agile+ practices and the organizational emphasis on flexibility facilitate needed corrective actions. OWN - Collective ownership makes everyone responsible for monitoring progress. GNT - Use of the Wall Gantt and its accompanying daily standup meeting means that slippages are made visible within 24 hours and can be corrected while they are small.



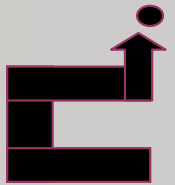
Supplier Agreement Management	<ol style="list-style-type: none"> 1. Establish Supplier Agreements 2. Satisfy Supplier Agreements 	FLX RSK OWN CON ARC INT CMG TST COD	In cases where there are third-party suppliers involved in a software development project, the Agile+ practices listed here involve interaction with the suppliers.
Measurement and Analysis	<ol style="list-style-type: none"> 1. Align Measurement and Analysis Activities 2. Provide Measurement Results 	ITR GNT	ITR - The frequent comparison of completed software to plan at each iteration provides accurate visibility into the real progress of the project. GNT - The daily discussions at the Wall Gantt standup meeting make highly visible the data needed for effective measurement.
Process and Product Quality Assurance	<ol style="list-style-type: none"> 1. Objectively Evaluate Processes and Work Products 2. Provide Objective Insight 	ITR	ITR - Iteration planning involves additional staff and management an extended, intense evaluation of project progress.
Configuration Management	<ol style="list-style-type: none"> 1. Establish Baselines 2. Track and Control Changes 3. Establish Integrity 	CMG	CMG - Robust Configuration Management is vital to Agile+
Requirements Development	<ol style="list-style-type: none"> 1. Develop Customer Requirements 2. Develop Product Requirements 3. Analyze and Validate Requirements 	CUS FLX BPA STO ACT	1-The customer representatives on the team represent the stakeholders. FLX - The flexibility of Agile+ allows for keeping the project in synch with requirements as they evolve. BPA - Business process analysis ties requirements to business need. STO - User stories are focused on being understandable to the customer. ACT - The user of story actors keeps focus on the characteristics of those who will actually use the software.



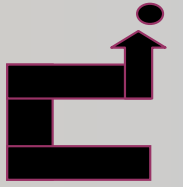
Organizational Process Definition	1. Establish organizational process assets		<i>Agile+ software development methodology while taking a different approach to managing it and meeting the organizational PAs</i>
Organizational Training	1. Establish an organizational training capability 2. Provide Necessary Training		
Integrated Project Management	1. Use the Project's Defined Process 2. Coordinate and Collaborate with Relevant Stakeholders 3. Use the Projects Shared Vision for IPPD Organize Integrated Teams for IPPD	CUS GEN OWN	CUS - The customer representatives on the team strengthens stakeholder involvement and helps maintain a shared vision. GEN - From the Agile+ 20 best practices, those appropriate for a given project are selected. OWN - Collective ownership facilitates shared vision.
Risk Management	1. Prepare for Risk Management 2. Identify and Analyze Risks 3. Mitigate Risks	RSK	RSK - The Risk-Based Situation Audit practice of Agile+, combined with project transparency and risk focused discussions during the daily stand-ups provides an intensely practical way to manage risk.
Integrated Teaming	1. Establish Team Composition 2. Govern Team Operation	CUS OWN COD PPG	CUS - To the extent possible, one or more customer representatives are an integral part of the project team. OWN - Collective ownership contributes greatly to the development of a high-performance team. COD - Since in Agile+ there is collective ownership of code, coding standards are essential to an integrated, collaborative approach to software development. PPG - Pair programming is utilized on high-risk, complex portions of the system, or as a training tool. GNT - Daily standup meetings at the Wall Gantt govern the day-to-day operation of the team.
Integrated Supplier Management	1. Analyze and Select Sources of Products 2. Coordinate Work with Suppliers	FLX RSK OWN CON ARC INT CMG TST COD	In cases where there are third-party suppliers involved in a software development project, the Agile+ practices listed here involve interaction with the suppliers.



Technical Solution	<ol style="list-style-type: none"> 1. Select Product-component Solutions 2. Develop the Design 3. Implement the Product Design 	ARC INT REF	<p>ARC - The architectural work at the beginning of an Agile+ project always takes a componentized approach, but also encompasses target hardware and OS, tool selection, and high-level technical and design decisions.</p> <p>INT - Clear definitions of both internal and external program interfaces are essential to Agile+ and a componentized architecture approach.</p> <p>REF - Refactoring produces a more robust, maintainable software solution and is made practical primarily by the Agile+ practice of Automated Contract and Regression Testing.</p> <p>ITR - An integral part of iteration planning is "days of design" where the detailed design for the current iteration is developed.</p>
Product Integration	<ol style="list-style-type: none"> 1. Prepare for product integration 2. Ensure Interface Compatibility 3. Assemble Product Components and Deliver the Product 	CON TST CRT CMG	<p>CON, TST, CRT, CMG - The practice of continuous integration is supported by relentless testing, automated contract testing, and configuration management such that the product is always in a stable deliverable, integrated state</p> <p>INT - Clear definitions of both internal and external program interfaces are essential to Agile+ and a componentized architecture approach.</p>
Verification	<ol style="list-style-type: none"> 1. Prepare for Verification 2. Perform Peer Reviews 	CUS TST	<p>CUS - The customer representatives on the team monitor the coupling of work products and requirements.</p> <p>TST - Relentless testing provides continuous assurance that work products are meeting requirements.</p> <p>GNT, PPG - Daily standup meetings and pair programming provide peer review.</p>
Validation	<ol style="list-style-type: none"> 1. Prepare for Validation 2. Validate Product or Product Components 	BPA	<p>BPA - Validating against BPA work products ensures that the product fulfills its intended use.</p>
Organizational Process Focus	<ol style="list-style-type: none"> 1. Determine process improvement opportunities 2. Plan and Implement Process Improvement Activities 		<p><i>These organizational PAs are not so much in the purview of a software development methodology itself, but rather how that methodology is institutionalized, improved, and taught.</i></p> <p><i>See Appendix 4 for a summary of AgileTek's approach to managing Agile+. Other organizations may adopt AgileTek's</i></p>



Decision Analysis and Resolution	1. Evaluate Alternatives	ITR GNT	ITR - Iteration planning affords a frequent opportunity to evaluate the direction of the project, examine alternatives, and make informed decisions. GNT - Many issues requiring the evaluation of alternatives arise at the daily Wall Gantt standup meeting and decisions are either made at that time or plans laid for how the decision will be made.
Organizational Environment for Integration	1. Provide IPPD Infrastructure 2. Manage People for Integration	FLX ITR COD	FLX - The flexibility of Agile+ removes barriers that otherwise could impede an effective integrated team. ITR - Iteration planning affords an expanded opportunity on a regularly scheduled basis for collaboration of the entire development/customer team. COD - Since in Agile+ there is collective ownership of code, coding standards are essential to an integrated, collaborative approach to software development.

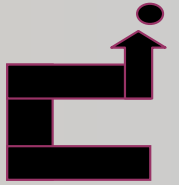


Misconceptions About Agile

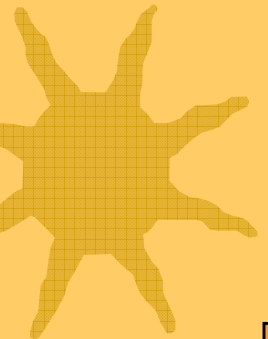
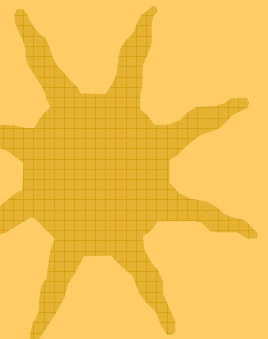
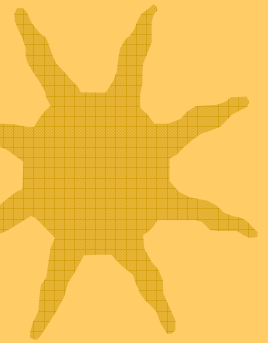
Agile Methods :

- Add Risk to a Project
- Don't Scale (for large long term projects)
- Won't Work in a Regulated Environment
- Incompatible With the Software CMM and CMMI

Conclusions



Using Agile+ to achieve CMMI Level 2 and 3 will provide several benefits that might not otherwise accrue:



THANK-YOU