

Software Engineering: A Practitioner's Approach,
6/e

Chapter 2

Process: A Generic View

copyright © 1996, 2001, 2005

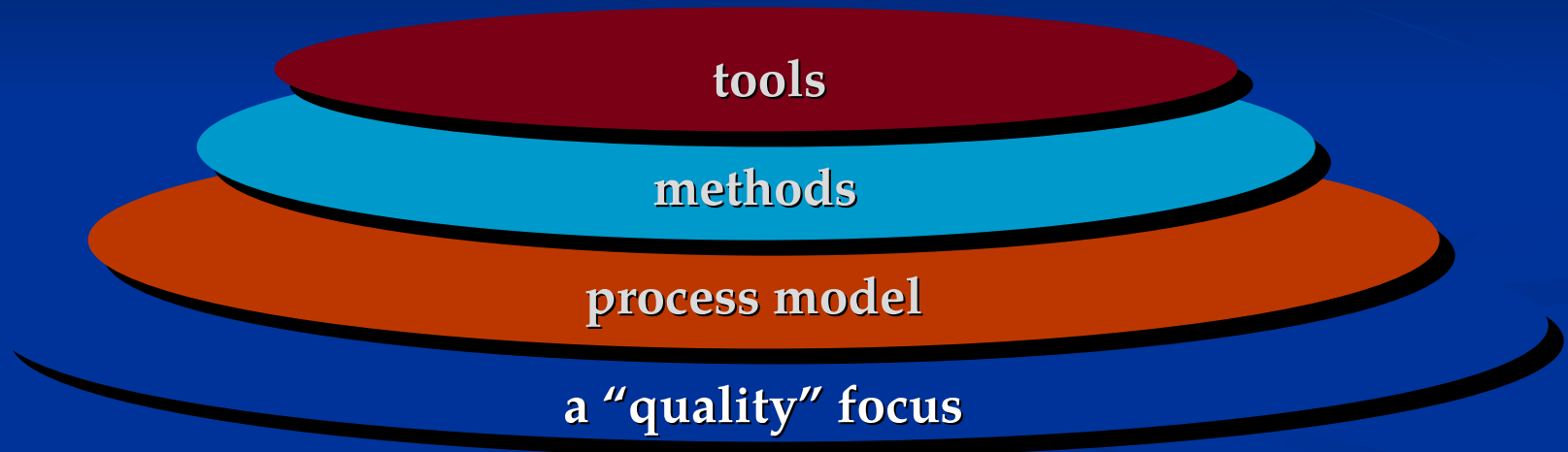
R.S. Pressman & Associates, Inc.

For University Use Only

May be reproduced ONLY for student use at the university level
when used in conjunction with *Software Engineering: A Practitioner's Approach*.
Any other reproduction or use is expressly prohibited.

A Layered Technology

Software Engineering



A Process Framework

Process framework

Framework activities

work tasks

work products

milestones & deliverables

QA checkpoints

Umbrella Activities

Framework Activities

- Communication
- Planning
- Modeling
 - Analysis of requirements
 - Design
- Construction
 - Code generation
 - Testing
- Deployment

Umbrella Activities

- Software project management
- Formal technical reviews
- Software quality assurance
- Software configuration management
- Work product preparation and production
- Reusability management
- Measurement
- Risk management

The Process Model: Adaptability

- the framework activities will always be applied on every project ... BUT
- the tasks (and degree of rigor) for each activity will vary based on:
 - the type of project
 - characteristics of the project
 - common sense judgment; concurrence of the project team

The CMMI

- The CMMI defines each process area in terms of “specific goals” and the “specific practices” required to achieve these goals.
- *Specific goals* establish the characteristics that must exist if the activities implied by a process area are to be effective.
- *Specific practices* refine a goal into a set of process-related activities.

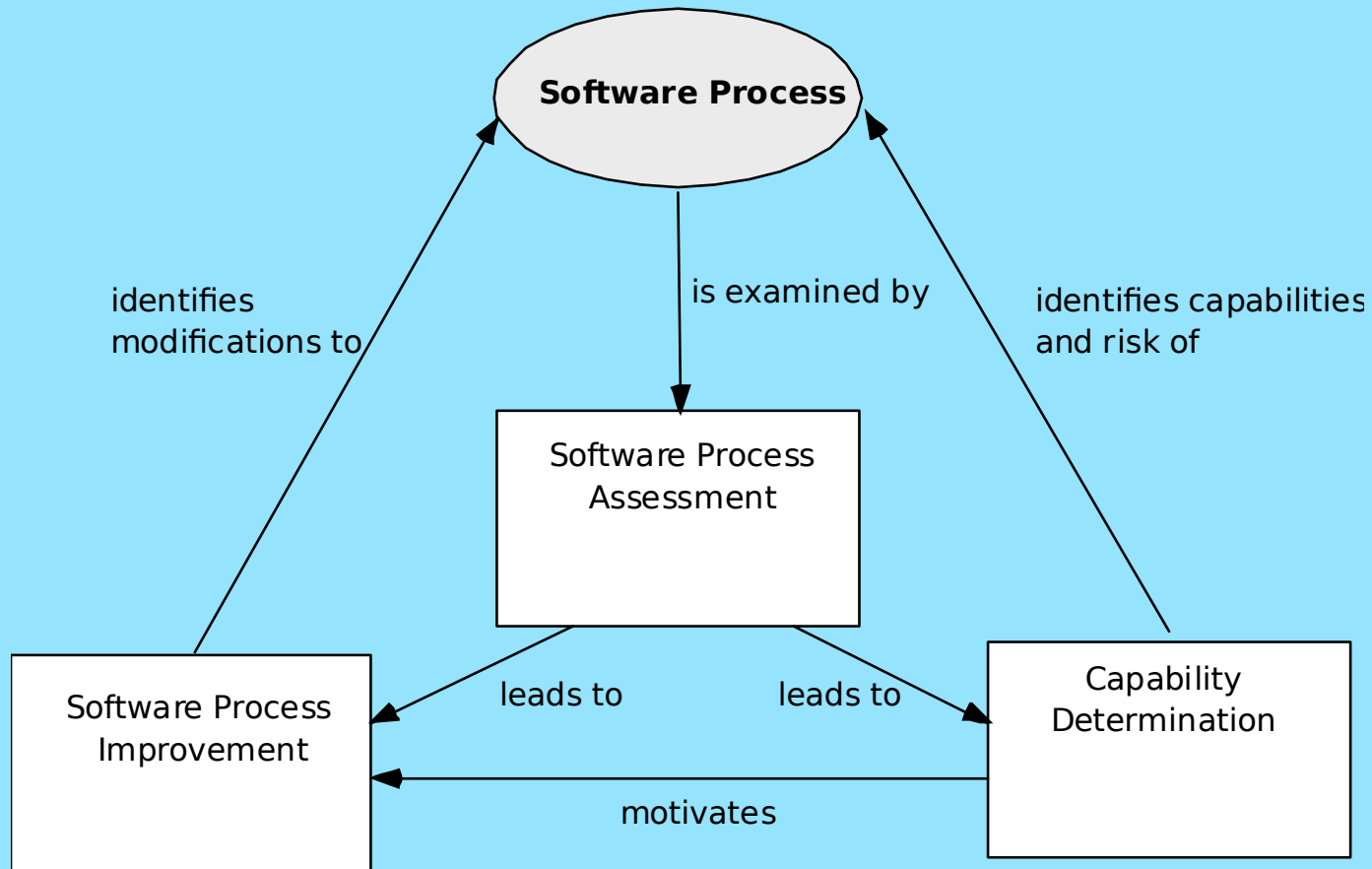
Process Patterns

- Process patterns define a set of activities, actions, work tasks, work products and/or related behaviors
- A template is used to define a pattern
- Typical examples:
 - Customer communication (a process activity)
 - Analysis (an action)
 - Requirements gathering (a process task)
 - Reviewing a work product (a process task)
 - Design model (a work product)

Process Assessment

- The process should be assessed to ensure that it meets a set of basic process criteria that have been shown to be essential for a successful software engineering.
- Many different assessment options are available:
 - SCAMPI
 - CBA IPI
 - SPICE
 - ISO 9001:2000

Assessment and Improvement



Personal Software Process (PSP)

- Every developer uses some process to build computer s/w.
- Process may be haphazard or adhoc.
- Watts Humphrey suggests that in order to change an ineffective personal process, an individual must move through 4 phases.
- PSP emphasizes *personal measurement of the work product produced & resultant quality of work product.*
- PSP makes practitioner responsible for Project Planning.

Personal Software Process (PSP)

- Recommends five framework activities:
 - Planning : resource , size, defect estimates.
 - High-level design: External specs for components & component design
 - High-level design review : uncover errors in design.
 - Development : code compiled tested.
 - Postmortem : using measures & metrics effectiveness of process determined.
- stresses the need for each software engineer to identify errors early and as important, to understand the types of errors

(PSP)

- Psp represents a disciplined , metrics –based approach to S.E.
- Problems.
- Intellectually challenging & demands a level of commitment by practitioners & managers.
- Training is lengthy & training costs are high

Team Software Process (TSP)

- Goal of TSP is to build a self directed project team that organizes itself to produce high quality software.
- Objectives of TSP
- Build self directed teams that plan & track their work 3-20 engg.
- Show managers how to coach & motivate their teams & sustain peak performance.
- Accelerate s/w process improvement by making CMM5
- Normal & expected.

Team Software Process (TSP)

- Each project is “launched” using a “script” that defines the tasks to be accomplished
- Teams are self-directed
- Measurement is encouraged
- Measures are analyzed with the intent of improving the team process

The Primary Goal of Any Software Process: *High Quality*

Remember:

High quality = project timeliness

Why?

Less rework!

■ History: The Agile Manifesto



On February 11-13, 2001, at The Lodge at Snowbird ski resort in the Wasatch mountains of Utah, seventeen people met to talk, ski, relax, and try to find common ground and of course, to eat. What emerged was the Agile Software Development Manifesto.

Representatives from Extreme Programming, SCRUM, DSDM, Adaptive Software Development, Crystal, Feature-Driven Development, Pragmatic Programming, and others sympathetic to the need for an alternative to documentation driven, heavyweight software development processes convened.

What is “Agility”?

- Effective (rapid and adaptive) response to change
- Effective communication among all stakeholders
- Drawing the customer onto the team
- Organizing a team so that it is in control of the work performed

Yielding ...

- Rapid, incremental delivery of software

Manifesto for Agile Software Development

■ We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to 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.

■

Kent Beck	James Grenning	Robert C. Martin
Mike Beedle	Jim Highsmith	Steve Mellor
Arie van Bennekum	Andrew Hunt	Ken Schwaber
Alistair Cockburn	Ron Jeffries	Jeff Sutherland
Ward Cunningham	Jon Kern	Dave Thomas
Martin Fowler	Brian Marick	

■ © 2001, the above authors
this declaration may be freely copied in any form,
but only in its entirety through this notice.

■ Principles behind the Agile Manifesto

1. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
2. Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
3. Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
4. Business people and developers must work together daily throughout the project.

5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.

9. Continuous attention to technical excellence and good design enhances agility.
10. Simplicity--the art of maximizing the amount of work not done--is essential.
11. The best architectures, requirements, and designs emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

An Agile Process

- Is driven by customer descriptions of what is required (scenarios)
- Recognizes that plans are short-lived
- Develops software iteratively with a heavy emphasis on construction activities
- Delivers multiple 'software increments'
- Adapts as changes occur

Assumptions:

- 1.It is difficult to predict in advance the requirements;**
- 2.Phases are interleaved.**
- 3.All the phases are not predictable.**

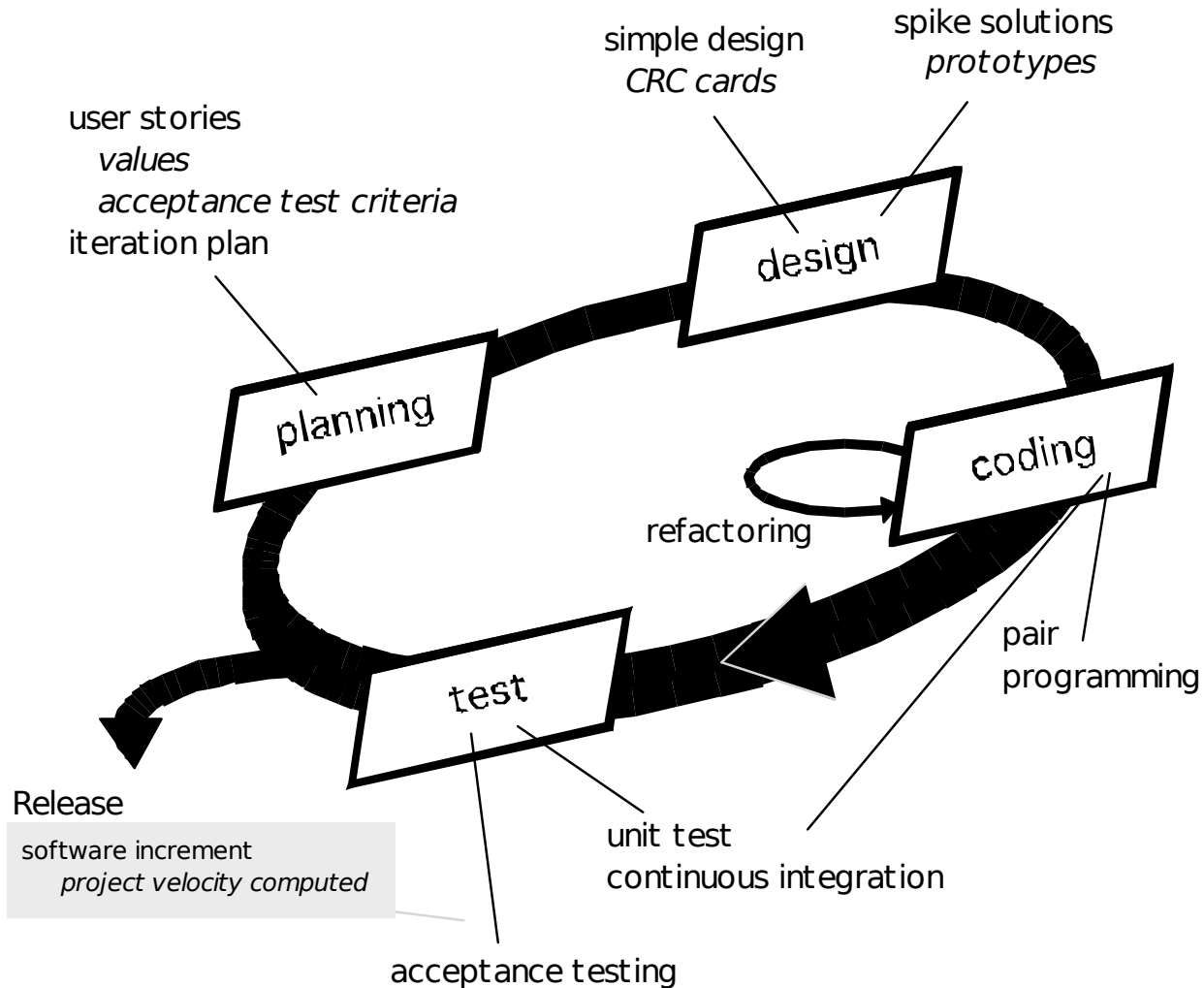
Extreme Programming (XP)

- The most widely used agile process, originally proposed by Kent Beck
- XP Planning
 - Begins with the creation of “user stories”
 - Agile team assesses each story and assigns a cost
 - Stories are grouped for a deliverable increment
 - A commitment is made on delivery date
 - After the first increment “project velocity” is used to help define subsequent delivery dates for other increments

Extreme Programming (XP)

- XP Design
 - Follows the KIS principle
 - Encourage the use of CRC cards (see Chapter 8)
 - For difficult design problems, suggests the creation of “spike solutions”—a design prototype
 - Encourages “refactoring”—an iterative refinement of the internal program design
- XP Coding
 - Recommends the construction of a unit test for a store *before* coding commences
 - Encourages “pair programming”
- XP Testing
 - All unit tests are executed daily
 - “Acceptance tests” are defined by the customer and executed to assess customer visible functionality

Extreme Programming (XP)



SECTION - I

- Q1) A) What is Software Engineering? What are the characteristics of software?
Explain in detail following software myths: [8]
- a) Management myths.
 - b) Practitioner's myths.
- B) Explain in detail the Unified Process indicating workflows and process phases. What are the advantages of iterative development? Compare Iterative development with Incremental delivery approach. [9]

OR

- Q2) A) What is Software Process? What are framework and umbrella activities? What is the importance of umbrella activities? Explain in detail all the process phases of waterfall process model and state merits/demerits of the same. [8]
- B) Explain in detail all the levels of Capability Maturity Model Integration with key process areas. What is Process Assessment and how it can be performed? Explain with suitable diagram. [9]

SECTION - I

1. a) "To transform a 4GT implementation into product, the developer must conduct thorough testing, meaningful documentation and other solution integration". Justify your answer.

6

b) What is prototyping in software development ? Provide example of software development project that would be amenable to prototyping. And provide example of software development project that would be more difficult to prototype.

10

OR

● a) Why the software maturity framework was developed ? Explain the key process areas of capability maturity model.

10

b) "Although the industry is moving towards component based assembly, most software continues to be custom built".

6

SECTION – I

1. A) What is Software Process Model ? Explain the Incremental process models. 8
- B) Explain in detail Process Patterns. What is the relationship between process and product ? 9

OR

2. A) What is Software Component ? What is Component Based Software Development ? What are the issues to be considered in selection and usage of components ? 8
- B) Explain in detail for project planning process area of CMMI: 9
- a) Specific Goals and Specific Practices.
 - b) Generic Goals and Generic Practices.

SECTION – I

1. A) What is meant by engineering the software ? What is meant by software evolution ? Explain the merits and demerits of incremental process model. 8
- B) What are the reasons to have a Software Process ? What are the issues addressed by Umbrella Activities in Layered Model of software engineering ? What are the levels of CMMI ? 9

OR

2. A) List and explain management and customer myths. Why a late project can not be placed back on schedule by merely adding people to the project teams ? What is the impact of scope change on project deliverables ? 8
- B) What is the objective of Personal Software Process (PSP) ? What are the activities of PSP model ? What are the goals of Team Software Process (TSP) model ? What are the conditions in which Rapid Application Development Model is preferred ? 9

SECTION - I

1. A) What is the importance of software engineering? What is meant by software evolution? What are management and customer myths? 8
- B) Explain in detail the following:
- a) Prototyping model
 - b) Concurrent development model. 9

OR

2. A) Explain in detail software engineering layers. What is the role of modeling and construction in software process models? 8
- B) Explain in detail Personal and Team Process models. 9

SECTION - I

1. a) Explain the generic process framework activities. 5
- b) Explain with neat diagram, the prototyping model for software development.
What are its drawbacks? 6
- c) Explain in detail the various phases of the unified process. 6

OR

2. a) Explain the umbrella activities which are applied throughout the software process. 5
- b) Explain with neat diagram, the spiral model for software development. 6
- c) Write short note on Rapid Application Development (RAD) model. 6