



CALIFORNIA STATE UNIVERSITY  
**FULLERTON**<sup>TM</sup>

---

COLLEGE OF ENGINEERING  
AND COMPUTER SCIENCE

# Advanced Software Process

Part I: Software Process Maturity

*1. A Software Maturity Framework*

Dave Garcia-Gomez

Faculty / Lecturer

Department of Computer Science

# Course Roadmap

## Part I: Software Process Maturity

- 1. A Software Maturity Framework
- 2. The Principles of Software Process Change
- 3. Software Process Assessment
- 4. The Initial Process

## Part II: The Repeatable Process

- 5. Managing Software Organizations
- 6. The Project Plan
- 7. Software Configuration Management (Part I)
- 8. Software Quality Assurance

## Part III: Defined Process

- 9. Software Standards
- 10. Software Inspections
- 11. Software Testing
- 12. Software Configuration Management (continued)
- 13. Defining the Software Process
- 14. The Software Engineering Process Group

## Part IV: The Managed Process

- 15. Data Gathering and Analysis
- 16. Managing Software Quality

## Part V: The Optimizing Process

- 17. Defect Prevention
- 18. Automating the Software Process
- 19. *Contracting for Software*
- 20. *Conclusion*

# Introduction

Companies have faced software problems for a long time.

- Open-ended requirements
- Uncontrolled change
- Arbitrary schedules
- Insufficient testing time
- Inadequate training
- Unmanaged system standards

# Introduction

- Scale of development ranges from individuals to teams to armies of developers
- Hiring talented people is beneficial, but then what?
- Different programming languages, special conventions, code change coordination contributes to chaos
- Organizations still have same problems

# Introduction

- Advanced technology tools would address the software issues.
- This is not only wrong, but is also dangerous.
- Diverts the need for better process management.
- Effective use of software technology is limited by ill-defined process, inconsistent implementation, and poor process management.

# Introduction

This course will help you deal with the following questions:

- How good is my current software process?
- What must I do to improve it?
- Where do I start?

# Introduction

This course will also provide you with the understanding of the following concepts:

CMM (the basis of this textbook)

ISO 12207 Software Lifecycle Processes

ISO 15504 Process Assessment (SPICE)

CMMI

SCAMPI

# Introduction

This course will also provide you with the understanding of the following concepts:

Agile Processes: SCRUM, XP

Unified Process

Object Oriented Modeling



# Introduction

- The SEI Software Capability Maturity Model (CMM) was built to provide guidance to the military services in selecting capable software contractors.
- The resulting method for evaluating their strengths and weaknesses has proved valuable for assessing other software organizations.

# Introduction

The textbook consists of 5 parts, each of which corresponds to a stage of the software process improvement.

- Part 1: Software Process Maturity (Initial)
- Part 2: The Repeatable Process
- Part 3: The Defined Process
- Part 4: The Managed Process
- Part 5: The Optimizing Process

# Introduction

## Part 1: Software Process Maturity (Initial)

Describes a **framework for software process management**, the use of this framework in process assessment, and the steps required to initiate effective software process change

# Introduction

## Part 2: The Repeatable Process

Outlines the actions needed to establish basic control over the software process.

These provide the stability needed for orderly and continuous process improvement.

# Introduction

## Part 3: The Defined Process

Describes how to specify the development process and the technical and managerial concepts needed to control it.

Software development is a surprise-prone business, and a defined process permits more orderly reaction to unanticipated events.

# Introduction

## Part 4: The Managed Process

The method of **quantitative software process control** are introduced.

At this stage data is gathered and analyzed to support **quantitative quality and process management**.

These are the tools that can guide us to a better understanding of our work, a more precise way to control our actions, and a truly informed basis for **sustained process improvement**.

# Introduction

## Part 5: The Optimizing Process

Presents the final stage of software process evolution; it shifts the focus from fixing problems to **preventing** them.

At this level software managers and professionals learn to **use quantitative process methods** to harness technology for continuing quality and productivity improvement.

# Introduction

Suggestion:

Outside of reading Chapter 1, also read Chapter 19 and Chapter 20, and the summaries of all chapters to get a good overview of the entire process.



# A Software Maturity Framework

The **software process** is the set of **tools**, **methods**, and **practices** we use to produce a software product (what, who, where.)

The objectives of **software process management** are to **produce products according to plan** while simultaneously improving the organization's **capability** to produce better products

# A Software Maturity Framework

The characteristics of a truly effective software process

- Fundamentally, it must be predictable.
- That is, cost estimates and schedule commitments must be met with reasonable consistency, and the resulting products should generally meet users' functional and quality expectations.

# A Software Maturity Framework

The objectives of software process management are to produce products according to plan while simultaneously improving the organization's capability to produce better products.

# A Software Maturity Framework

The basic principles are those of **statistical process control**, which have been used successfully in many fields.

A process is said to be **stable or under statistical control** if its future performance is **predictable within established statistical limits**.

When a process is under **statistical control**, repeating the work in roughly the same way will produce roughly the same result.

# A Software Maturity Framework

- To obtain **consistently better results**, it is thus necessary to **improve the process**.
- If the process is not under **statistical control**, sustained progress is not possible until it is.
- The basic principle behind **statistical control** is **measurement**.

# A Software Maturity Framework

There are several factors to consider in measuring the process.

- The numbers must properly represent the process being controlled.
- The numbers must be sufficiently well defined and verified to provide a reliable basis for action.
- The mere act of measuring human processes changes them.

It is thus essential to limit the measurements to those with a predefined use.

# Software Process Improvement

Important first step in addressing problems in the software process, is to treat the entire software task as a process that can be controlled, measured, and improved.

# Software Process Improvement

The improvement of software development organizations follows **six steps**:

1. Understand the **current status** of the development process.
2. Develop a **vision** of the desired process.
3. Establish a **list of required process improvement actions** in order of priority.
4. Produce a **plan** to accomplish the required actions.
5. Commit the resources to **execute the plan**.
6. **Start over** at step 1.



# Process Maturity Levels

## Five Levels of Process Maturity

### 1. Initial

Basic Management Control

### 2. Repeatable

Process Definition

### 3. Defined

Process Measurement

### 4. Managed

Process Control

### 5. Optimizing

# Process Maturity Levels

## CMM

- Initial
- Repeatable
- Defined
- Managed
- Optimizing

## CMMI

- Initial
- Managed
- Defined
- Quantitatively  
Managed
- Optimizing

# Process Maturity Levels

## Initial (Level 1)

- Until the process is under statistical control, orderly progress in process improvement is not possible.
- While there are many degrees of statistical control, the first step is to achieve rudimentary predictability of schedules and costs.

# Process Maturity Levels

## Repeatable (Level 2)

- The organization has achieved a **stable process with a repeatable level of statistical control** by initiating rigorous project management of commitments, cost, schedules, and changes.

# Process Maturity Levels

## Defined (Level 3)

- The organization has **defined the process** as a basis for consistent implementation and better understanding.
- At this point advanced technology can usefully be introduced.

# Process Maturity Levels

## Managed (Level 4)

- The organization has initiated **comprehensive process measurements and analysis**.
- This is when **the most significant quality improvements begin**.

# Process Maturity Levels

## Optimizing (Level 5)

- The organization now has a foundation for continuing improvement and optimization of the process.

# Process Maturity Levels

Process maturity levels have been selected because they:

- Reasonably represent the **actual historical phases of evolutionary improvement** of real SW organizations
- Represent a **measure of improvement** that is reasonable to achieve from the prior level
- Suggest **interim improvement goals and progress measures**
- Make obvious a **set of immediate improvement priorities**, once an organization's status in this framework is known



# The Initial Process (Level 1)

- Ad hoc
- Often even chaotic
- At this stage the organization typically operate without formalized procedures, cost estimates, and project plans.
- Tools are neither well integrated with the process nor uniformly applied.
- Change control is lax, and there is little senior management exposure or understanding of the problems and issues.

# The Initial Process (Level 1)

- While organizations at this level may have formal procedures for planning and tracking their work, there is **no management mechanism** to ensure that they are used.
- The best test is to observe how such an organization behaves in a **crisis**.

If it abandons established procedures and essentially reverts to coding and testing, it is likely to be at the **Initial Process Level**.

# The Initial Process (Level 1)

Organizations at the Initial Process Level can improve their performance by instituting basic project controls.

- Project management
- Management oversight
- Quality assurance
- Change control

# The Initial Process (Level 1)

## Project Management

- The fundamental role of a project management system is **to ensure effective control of commitments.**

Adequate preparation, clear responsibility, a public declaration, a dedication to performance

- Starts with **an understanding of the job's magnitude**

In any projects, **a plan** must be developed **to determine the best schedule and the anticipated resources required.**

# The Initial Process (Level 1)

## Management Oversight

- A suitably disciplined SW development organization must have **senior management oversight**.
  - **Review and approval** of all major development plans prior to their official commitment
  - **A quarterly review** should be conducted of facility-wide process compliance, installed quality performance, schedule tracking, cost trends, computing service, and quality and productivity goals by project.
  - The lack of such reviews typically results in uneven and generally inadequate implementation of the process as well as frequent over-commitments and cost surprise.

# The Initial Process (Level 1)

## Quality Assurance

- A quality assurance group is charged with assuring management that SW work is done the way it is supposed to be done.
- To be effective, the assurance organization must have an independent reporting line to senior management and sufficient resources to monitor performance of all key planning, implementation, and verification activities.
- This generally requires an organization of about 3 percent to 6 percent the size of the SW organization.

# The Initial Process (Level 1)

## Change Control

- Change control for SW is fundamental to business and financial control as well as to technical stability.
- To develop quality SW on a predictable schedule, **requirements must be established and maintained with reasonable stability** throughout the development cycle.
- Requirement changes; design and code changes
- If changes are not controlled, then orderly design, implementation, and test is impossible and no quality plan can be effective.

# The Repeatable Process (Level 2)

The **Repeatable Process** provides control over the way the organization establishes its plans and commitment.

Organizations at this level face major risks when they are presented with new challenges.

- New tools and methods will affect the process.
- When the organization must develop a new kind of product, it is entering new territory.
- Major organizational changes can also be highly disruptive.



# The Repeatable Process (Level 2)

Key actions required to advance from the Repeatable to the next stage, the Defined Process, are to

- Establish a process group (EPG, SEPG)
- Establish a software development process architecture (or development life cycle)
- Introduce a family of SE methods and technologies

# The Repeatable Process (Level 2)

## Establish a Process Group

- A process group is a **technical resource that focuses exclusively on improving the SW process.** (EPG, SEPG)
- In SW organizations at early maturity levels, all the people are generally devoted to product work.
- Until some people are given full-time assignments to work on the process, little orderly progress can be made in improving it.

# The Repeatable Process (Level 2)

## Establish a SW development process architecture, or development life cycle

- ... that describes the **technical and management activities** required for proper execution of the development process
- The **process architecture** is a **structural decomposition of the development life cycle into tasks** **architecture** is a, each of which has a defined set of prerequisites, functional descriptions, verification procedures, and task completion specifications.

The decomposition continues until each defined task is performed by an individual or single management unit.

# The Repeatable Process (Level 2)

## Introduce a Family of SW Engineering Methods and Technologies

- These include design and code inspections, formal design methods, library control systems, and comprehensive testing methods.
- Prototyping should also be considered, together with the adoption of modern implementation languages.

# The Defined Process (Level 3)

With the **Defined Process**, the organization has achieved the foundations for major and continuing progress.

- For example, the software team, when faced with a crisis, **will likely continue to use the process that has been defined.**
- The foundation has now been established for **examining the process and deciding how to improve it.**

# The Defined Process (Level 3)

The key steps required to advance from the Defined Process to the next level:

- Establish a **minimum basic set of process measurements** to identify the quality and cost parameters of each process step.
- Establish a **process database and the resources** to manage and maintain it.
- Provide sufficient process resources **to gather and maintain this process data** and to advise project members on its use.
- Assess the **relative quality of each product** and inform management where **quality targets** are not being met.

# The Managed Process (Level 4)

In advancing from the previous levels to the **Managed Process**, organizations should expect to make **substantial quality improvement**.

- The greatest problem with the Managed Process is the **cost of gathering data**.
- There are **an enormous number of potentially valuable measures** of the software process, but such data is expensive to gather and to maintain.

# The Managed Process (Level 4)

The two fundamental requirements for advancing from the Managed Process to the next level are:

- Support automatic gathering of process data.
- Use process data both to analyze and to modify the process to prevent problems and improve efficiency.



# The Optimizing Process (Level 5)

In varying degree, **process optimization** goes on at all levels of process maturity.

- With the step from the Managed Process to the Optimizing Process, however, there is **a paradigm shift**.
- Up to this point SW development managers have largely **focused on their products** and will typically gather and analyze only data that directly relates to **product improvement**.

# The Optimizing Process (Level 5)

- In the Optimizing Process, the data is available to tune the process itself.
- With a little experience, management will soon see that process optimization can produce major quality and productivity benefits.
- With the Optimizing Process, the organization has the means to identify the weakest elements of the process and to fix them.

# People in the Optimizing Process

Clearly any SW process is dependent on the quality of the people who implement it.

The Optimizing Process enhances the talent of quality people in several ways.

- It helps managers understand where help is needed and how best to provide people with the support they require.
- It lets the professionals communicate in concise, quantitative terms.
- It provides a framework for the professionals to understand their work performance and to see how to improve it.

# The Need for the Optimizing Process

- Unless we dramatically improve our error rates, the **greater volume of code** will mean increased risk of error.
- The **complexity of our systems** is increasing, which will make the systems progressively more **difficult to test**.
- As well as being a management issue, **quality** is an economic one.

It is always possible to do more tests, but it costs time and money to do so.

# The Need for the Optimizing Process

- It is only with the Optimizing Process that the data is available to understand the costs and benefits of such work.
- The Optimizing Process provides the foundation for significant advances in SW quality and simultaneous improvements in productivity.

# The Need for the Optimizing Process

Transition from Level 1 to Level 2 or from Level 2 to Level 3 take from one to three years, even with a dedicated management commitment to process improvement [Humphrey 1989].

# Process Levels Comparison

	<b>CMM Maturity</b>	<b>CMMI Maturity</b>	<b>CMMI v1.2 Capability</b>	<b>CMMI v1.3 Capability</b>	<b>ISO 15504 Capability</b>
Level 5	Optimizing	Optimizing	Optimizing	-	Optimizing
Level 4	Managed	Quantitatively Managed	Quantitatively Managed	-	Predictable
Level 3	Defined	Defined	Defined	Defined	Established
Level 2	Repeatable	Managed	Managed	Managed	Managed
Level 1	Initial	Initial	Performed	Performed	Performed
Level 0	-	-	Incomplete	Incomplete	Incomplete

# References

Humphrey, Watts S., *Managing the Software Process*, The SEI Series in Software Engineering, Addison-Wesley, 1989. (29th Printing, May 2003) (ISBN 0-201-18095-2)