# SW Project Management

Alka Jarvis

## Session: 1
## The Problem

- “Software crisis” of 1960s and 1970s was due to high profile software project failures
- Larger projects were tackled with techniques developed for much smaller projects
- Organizations were unaware of specific techniques needed for SW project management
- Good project management may not guarantee 100% success but poor project management will surely lead the project into a grand failure!

## The Difference

SW Projects are different than other projects:

- The product is intangible: It is easy to claim a software project is 90% complete even if there is no visible outcome
- SW engineering is a new discipline and we don't have clear understanding on how to engineer large scale projects
- Most large projects are unique and it is hard to apply the experience from one to another and reuse techniques

## Successful SW Project

Combination of:

- Skills
- Collaboration and teamwork techniques
- Experience: SW Engineering
- SW Quality
- Estimation techniques
- Risk evaluation
- ▪ Mitigation and Remedial actions

## SW Project Complexity

- Communication with customers and end-users
- Designing and producing different artifacts
- Working with different technologies
- Installation and maintenance
- Providing Customer support
- Evaluating training needs
- Envisioning potential upgrades and negotiating with customers regarding the upgrades

# Important Issues of SW Project Management

SW Project Management

Process Improvement

Requirement Engineering

Project Planning

SW Testing

Risk Assessment

SW Quality Assurance

SW Metrics

Standards

SW Change Management

SW CM

## SW Project Planning

**Develop realistic plan to understand the type and number of resources required**

**Various Plans**

- **Software Project Management Plan**
- **Software Development Plan: describes how the system will be developed**
- **Quality Assurance Plan: Specifies the quality procedures and Standards to be used**
- **Validation Plan: defines how the system will be validated (tested)**
- **Configuration Management Plan: Defines how the system will be configured and installed**
- **Staff Development Plan: Describes how the skills of the participants will be developed**

## SW Project Management Plan: TOC

- Introduction
  - Project Overview
  - Project Deliverables
  - Evolution of SPMP: describe how this plan will be disseminated and put under change control
  - Reference Material: Provide list of other documents referenced in this Plan
  - Definitions and Acronyms

## SW Project Management Plan: TOC

- Project Organization
  - Specify lifecycle model to be used: Process model to include roles, activities, entry/exit criteria, product development, product release, project end
- Organizational Structure
  - Describe how the project relates to other existing projects and rest of the organization
  - Organizational Interfaces: customer, subcontractors, SQA, SW CM
  - Project Responsibilities: Identify each major activity and individuals responsible

## SW Project Management Plan: TOC… cont.

- Managerial Process
  - Management Objectives and Priorities (Project Dimensions: cost, schedule, scope: fixed/constrained/flexible)
  - Assumptions, dependencies and constraints
  - Risk Management
  - Monitoring and controlling methods
  - Staffing Approach: types of skills required for the project, how appropriate personnel will be recruited and trained

# SW Project Management Plan: TOC… cont.

- Technical Process
  - Methods, tools and techniques: identify the computing system(s), standards, policies, procedures, programming language
  - Software Documentation: Specify work products to be built for this project and types of peer reviews to be held. Identify naming conventions to be used or specific document formats to be used
  - Software Requirements Specifications (SRS): Each requirement is defined
  - Software Design Description (SDD): The SDD describes major components of the SW design including databases and internal interfaces

Derived from IEEE Standard for SW Management Plan

## SW Project Management Plan: TOC… cont.

- Technical Process...cont.
  - SW Test Plan: Describe the methods to be used for testing all levels of development and integration, including test cases and test results
  - User Documentation: Describe how user documentation will be planned and developed
  - Work Packages, schedule and Budget: Describe work packages, dependency, relationships, resource requirement, allocation of budget and project schedule
  - Dependencies: Specify relations among work packages to identify interdependencies among them

Derived from IEEE Standard for SW Management Plan

## SW Development Plan

- Specifies how the work will be done
- Important not to over-estimate your team's ability
- Structure of Development Plan
  - Introduction: introduction of project, references the requirement specification
  - Project Organization: people and their roles
  - Risk Analysis: what are the key risks to the project?
  - HW & SW Resources: What resources will be required for the project and when?
  - WBS: Individual activities, milestones, deliverables and dependencies
  - Project Schedule: actual time required and allocation of dates
  - Measurements: mechanisms to monitor progress

## SW Quality Assurance Plan

- Objective
- Who will be impacted by this system?
- Resources responsible to carry out the activities that will contribute to the overall quality of the system
- Documentation: list all the documents that will be generated and what steps will be taken to ensure quality and how the documents will be maintained
- Deliverables of each phase
- Acceptance Criteria

## SW Quality Assurance Plan… cont.

- Delivery: Method and schedule
- Maintenance: who, when and what!
- Configuration Management
- Changes
- Quality records
- Metrics
- Training

## Session: 2 Elements to consider during SW Quality Assurance Planning

Establish processes to ensure the product is built right the first time:

- Reviews
- Entrance exit criteria
- Establishing quality targets for each release
- If targets are not met, conducting root cause analysis
- Conducting post project assessment of each system
- Establishing in-process metrics (leading indicator)

## Elements to consider during SW Quality Assurance Planning… cont.

- Documentation standards
  - ✓Format and contents
  - ✓What type of documents must be developed
- Coding standards
  - ✓ Variable naming conventions
  - ✓ Comment standards
  - ✓ Testing conventions

# Verification Plan

- What  is testing:
  - Process of identifying variance between actual and expected results
  - Destructive, not constructive process
  - Testing validates that a product meets requirements

- Effective Testing Techniques
  - Independent testing
  - Early test planning
  - Testing done over life cycle

## Verification Plan… cont.

- Unit test:
  - Performed by programmer
  - Testing of individual modules of code and then integrated system
  - Identify discrepancies between the performance of program logic

- System test:
  - Final, comprehensive execution of system
  - Performed by independent testers
  - Identify discrepancies between system performance and requirements
  - Verification of system interfaces
  - Last stage of integration testing

## Verification Plan… cont.

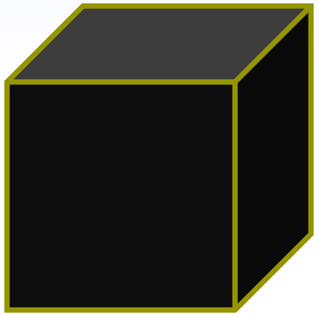
- Acceptance test:
  - Performed by users
  - Identify discrepancies between system execution and client expectations
  - Based on requirements

- Parallel test:
  - Performed by users or independent testers
  - High degree of consistency between older and new systems
  - Validate environmental discrepancies

# Verification Plan… cont.

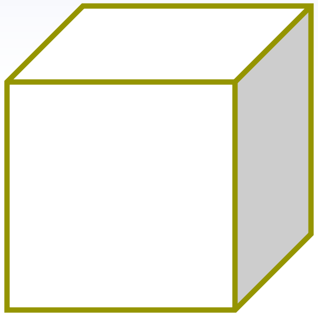
## Testing Design

### Black Box



- System oriented
- Different system tests conducted

### White Box



- Logic oriented
- Testing of paths

## Verification Plan… cont.

- **Testing Approach:**
  - **Build strategy**
  - **Test data**
  - **Test tools**
  - **Work plan and responsibilities**
  - **Procedures**

- **Test Case Design:**
  - **Define detailed input and expected results**
  - **Identify few test cases which uncover most flows**

# Verification Plan… cont.

## Testing Case Design

| | | |
|---|---|---|
| **Destructive**<br>• Is able to detect errors | → | **Clear expectation** |
| **Recordable**<br>• Easy to understand<br>• Can be measured | → | **Traceable results** |
| **Controllable**<br>• Follow predefined paths<br>• Use predefined input | → | **Consistency** |
| **Repeatable**<br>• Gives you same results each time | → | **Error detection** |

# Verification Plan… cont.

## Changes During Testing

- Changes must be centrally controlled
- Logs used to track changes and monitor resolution
- Regression testing planned to ensure positive/negative “domino” effect

## Configuration Management Plan

- Manage source and executable libraries
- Code Management: maintain program versions
- Track and control changes:
  - Who worked on the latest code?
  - Reason for the changes
  - Date the changes were made
  - Prevent unauthorized changes

## Staff Development Plan

- Gap analysis of current skill-set and what is required for the project
- Mentor program
- Video-based training
- Class-room training
- Seminars
- Web-based, self taught classes
- Verification of what was learned and whether it can be implemented

## Key SW Development Activities

- Requirements & prototyping
- Design & reviews
- Code & reviews
- Change management
- Configuration Management
- Testing
- User Documentation
- Maintenance

For each of the key activity, Work Breakdown Structure needs to be identified for estimation

## Most Common Work Breakdown Structures

- Work Packages
  - Large sections of work, typically of 12 month duration
  - May include multiple concurrent activities
  - Independent of other activities but may depend on or feed into other activities
  - Generally allocated to a single team
- Task
  - A much smaller piece of work, a part of a work package
  - Usually 3-6 person months effort
  - May be dependent on other concurrent activities
  - Mostly allocated to a single person

## Most Common Work Breakdown Structures… cont.

- Deliverable
  - Output of the project that can meaningfully be assessed
  - Examples: report, code
  - Deliverables are indicators of progress
- Milestone
  - A point at which progress on the project may be assessed
  - A major turning point in the project, for example: delivery of requirement document; delivery of design document
  - May be dependent on other concurrent activities

## Most Common Work Breakdown Structures… cont.

- For each work package and task document:
  - Brief description
  - Start date and end date
  - Total person months effort
  - Pre-requisite work package or tasks
  - Dependent work packages or tasks
  - The individual who is responsible for completing it

# Session: 3
# Estimation Plan

## Soft Costs versus Hard Costs

- Soft Cost
  - Example: planning and coordination of a software project
  - Good-will of customers
  - Adherence to a regulatory standard

- Hard Cost
  - Servers
  - Computers
  - Testing Tools

# Estimation Plan… cont.

## Estimation Method

- Resource Input
  - Ask resources identified as being required to complete work packages to provide detailed estimate by tasks and milestones
  - Use “% complete metric” which will force consideration of everything involved in a work package
  - When one or more resource is assigned to an activity, collect estimate independently
  - If estimates vary substantially, hold meeting between project manager and all resources to reach an agreement on final estimate

## Estimation Plan… cont.

### Per Diem Rate

- Per diem is Latin for "for a day"
- Calculate a per diem rate by dividing the full-time salary by 2080, the number of hours in a 52-week year of 40 hours per week
- Include benefits such as healthcare, vacation pay, sick leave, matching 401K

## Estimation Plan… cont.

### Estimation Method

- Organizational Project History Data
  - Review data from past projects that are most relevant to fine-tune the estimates for the new project

- Perform cost estimation by:
  - multiplying the amount of work expected by the hourly rate for the resources who are selected to perform work
  - multiplied by the % of participation that each resource expects to make toward the activity

## Estimation Plan… cont.

### Estimation Method

- Roll the estimates of each task (activity) for a work package

- The highest-level activity in the WBS, attach:
  - Schedule,
  - Resources
  - Cost estimates

The above will reflect the schedule and cost estimates for the entire project

## Re-Estimation Strategy

- When re-estimation is required:
  - Resource input:
    - The amount of work remaining
    - Estimate on breakdown of work remaining
    - Restate milestones if necessary
    - After re-estimating, if schedule is adversely affected (+/- 10%), use project history data to determine if additional resources will assist
    - Review data from past projects that are most relevant to fine-tune the estimates for the new project

## Re-Estimation Schedule

- Determine how often and when re-estimation will be addressed
  - Review resource needs: affected resources, managers of the affected resources
  - Cost:  does the increase in costs exceed the project budget?  Do you have to re-calculate ROI?
  - Schedule: Will it affect the overall deadline established before?  What will be the impact?

## Errors In Estimation Due To Historical Data

- Failure to include all the activities that were performed
  - Analysis of lessons learned from past project
  - Analysis of re-usable code from the past
  - Time taken to determine regression test strategy
  - Etc……..

- Failure to include all different workers who worked on the project
  - Program Managers Hours
  - Work performed by senior management
  - Technical work where IT department was consulted
  - Etc……….

## Categories of Estimation Errors

- Creeping requirements
- Failure to identify critical paths
- Project size
- Required staffing
- Technology adjustment: failures to calculate effects of various languages, methodologies and tools
- Scope related
- User errors
- Development errors

## Estimation Tools

### Manual Estimation

- Function Points
- LOC

### Automated Estimation

- SLIM
- Check point
- COCOMO
- SIZE Plus
- Project Bridge and many others

## Manual Estimation

LINE OF CODE MEASURE

- Empty lines
- Comments/statements
- Source lines
- Reused lines
- Headings

## Manual Estimation… cont.

### FUNCTION POINTS

- Logical Internal files
  Data stored & maintained by system
- External Interface Files
  Data passed or shared with other systems
- External Inputs
  Transactions, additions, changes
- External Outputs
  Reports
- External Inquiries: No updates

## Benefits of Function Points

The measure can be used for:

- measuring productivity
- measuring software quality
- any programming language
- combination of any language
- any industry sector software

## Automated Estimation

Many elements for estimation are derived from similar projects

1. Nature of the Project: new, enhancement or a maintenance
2. Scope of the Project: is it a small module or a large system
3. Class of the Project: is the SW for internal use or external

## Automated Estimation…cont.

4. **Project Type:** is it an information system, embedded software or systems software

**System Software:** any computer software which manages and controls computer hardware for the application software to perform a task, eg. Operating system for Mac or Linus

**Embedded System:** An embedded system has operating system on a "chip" embedded into the system and does not rely on having a hard disk with the operating system on it

**Information System**: Combination of people, machines and activities aimed at the gathering and processing of data to supply the information needed by people

## Automated Estimation… cont.

5. What is the staff salaries
6. How does the work week look like: 40 hours? 20 hours?
7. Size of the project: Source Code LOC
8. Language
9. Start Date

Once these information is plugged in, you can change the tool from input to estimation mode for initial cost estimates

## Automated Estimation… cont.

Class of the Project

Internal use:
single location, multiple location, to be developed by a contractor, etc.

External use:
bundled with hardware, developed under military regulations, built under some other regulatory requirements etc.

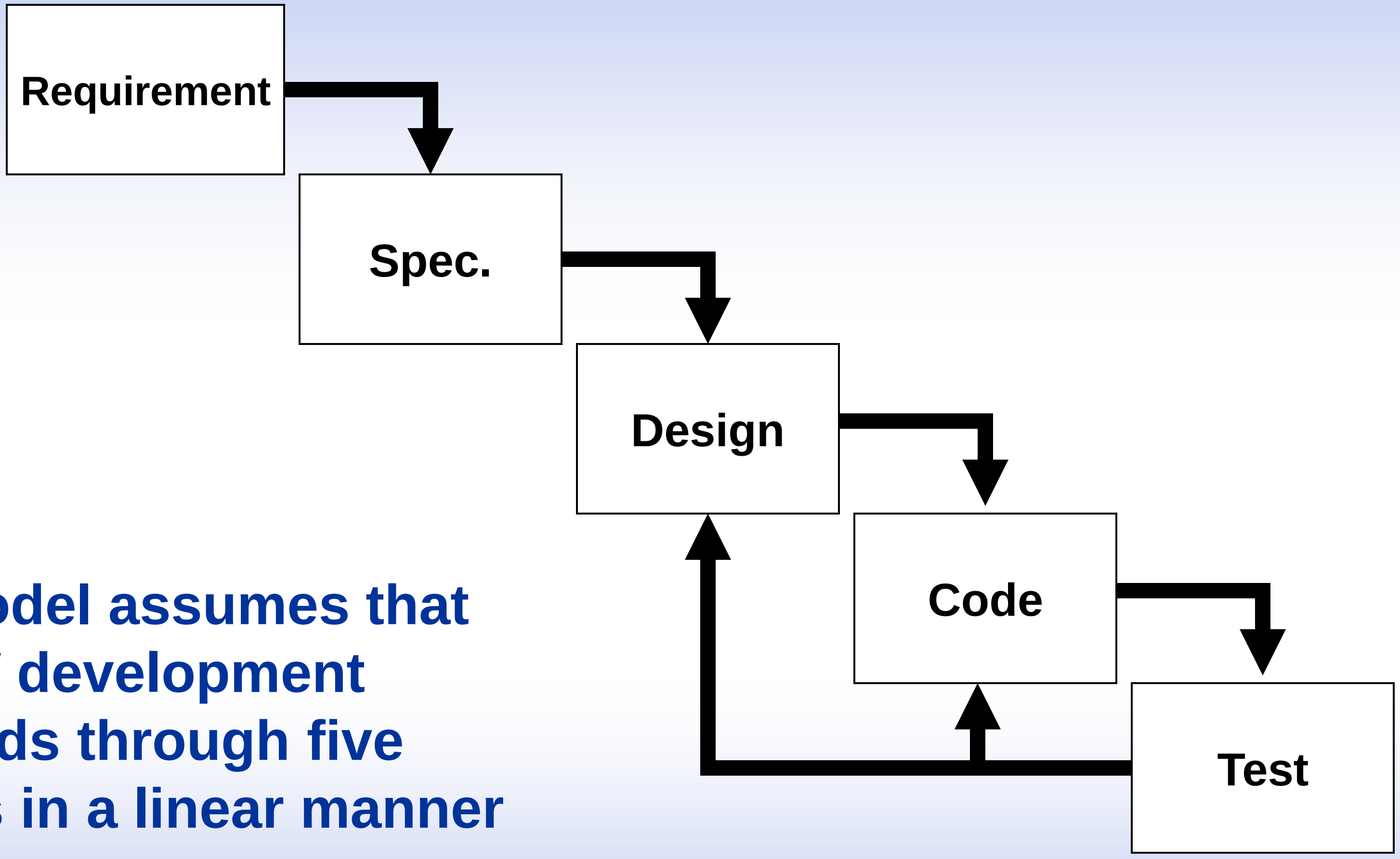
# Resource Requirement Planning

| Function | Hours | Dates | Phase | Qty | Hourly Rate |
|---|---|---|---|---|---|
| Program Manager | 2000 | 1/15/2012-3/15/2013 | All | 1 | $250-$300 |
| Req Analyst | 150 | 3/1/2012-4/14/2012 | Requirements | 1 | $125-150 |
| SW Architect | 300 | 3/1/2012-4/30/2012 | Design | 1 | $165-$250 |
| Programmer (Lead) | 800 | 4/1/2012-10/30/2012 | Code/Test | 1 | $150-$200 |
| Testers (system) | 530 | 4/1/2012-2/15/2013 | Req/Test | 3 | $100-$150 |

## Session: 4
## SW Development Process

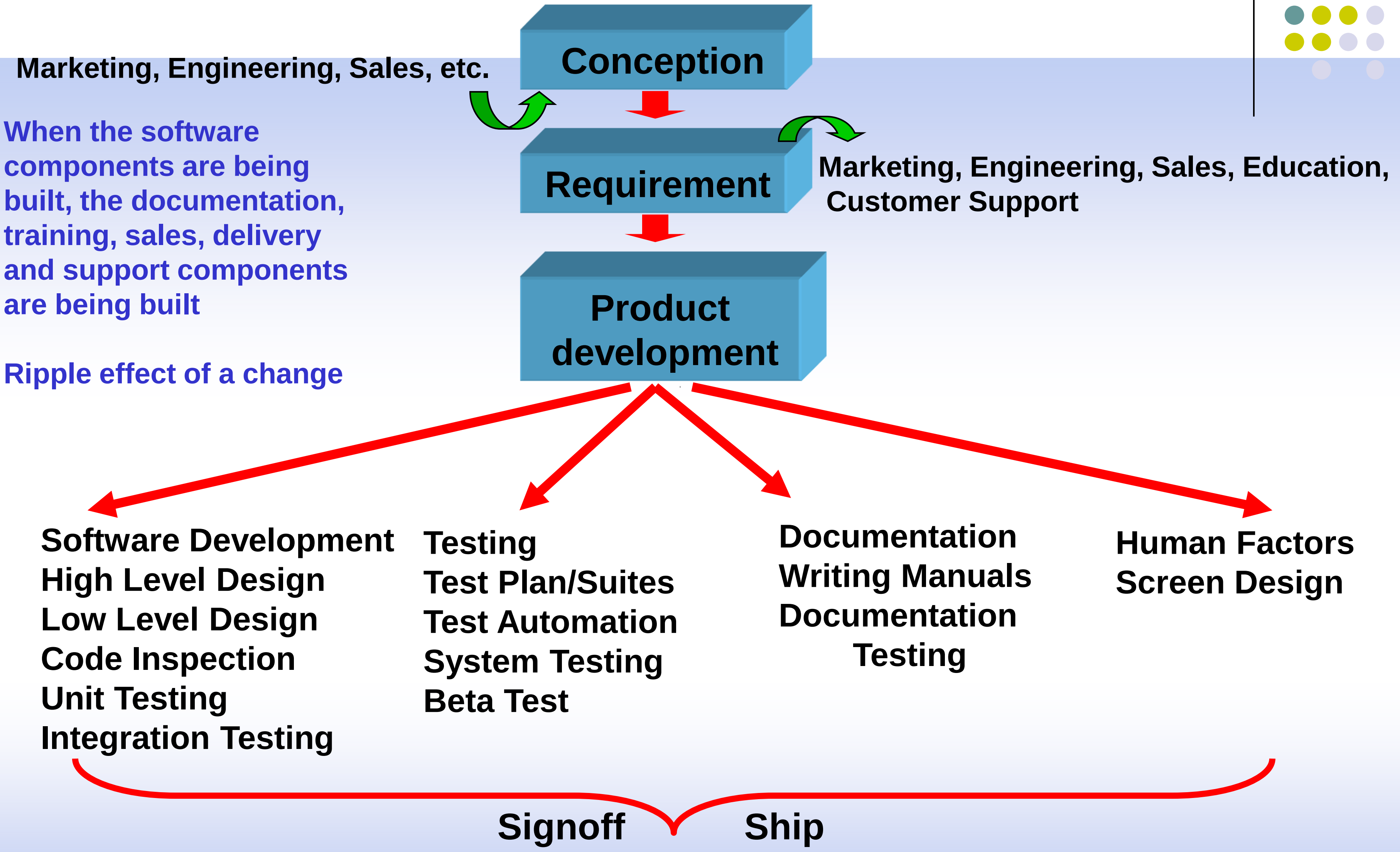
- Purpose of SDLC
  - Easy to understand WBS of each activity
  - Consistent templates
  - All employees follow the same process
  - Consistency in measurements of the project quality
  - Evaluation of team deliverable

## Model 1: Waterfall



The model assumes that the SW development proceeds through five phases in a linear manner

# Model 2: Parallel Product Development



When the software components are being built, the documentation, training, sales, delivery and support components are being built

Ripple effect of a change

## Model 3: Spiral

## Model 4: Agile

- Breaks projects into small increments with minimal planning, and do not emphasize on long-term planning. Iterations are short time frames that may last from one to four week

- Agile is  iterative and incremental (evolutionary) approach which is performed in a collaborative manner

- Each iteration involves a team working through a full software development cycle including planning, requirements analysis, design, coding, unit testing, and acceptance testing when a working product is demonstrated to stakeholders

## Agile…. Cont.

- It helps teams respond to the unpredictability of building software through incremental, iterative work cadences, known as sprints

- At the end of each sprint cycle, all completed and differed items are reviewed.  In some organizations, all members review the last cycle, identifying things that worked well/things that didn't

- The methodology minimizes overall risk and allows the project to adapt to changes quickly. Stakeholders produce documentation as required

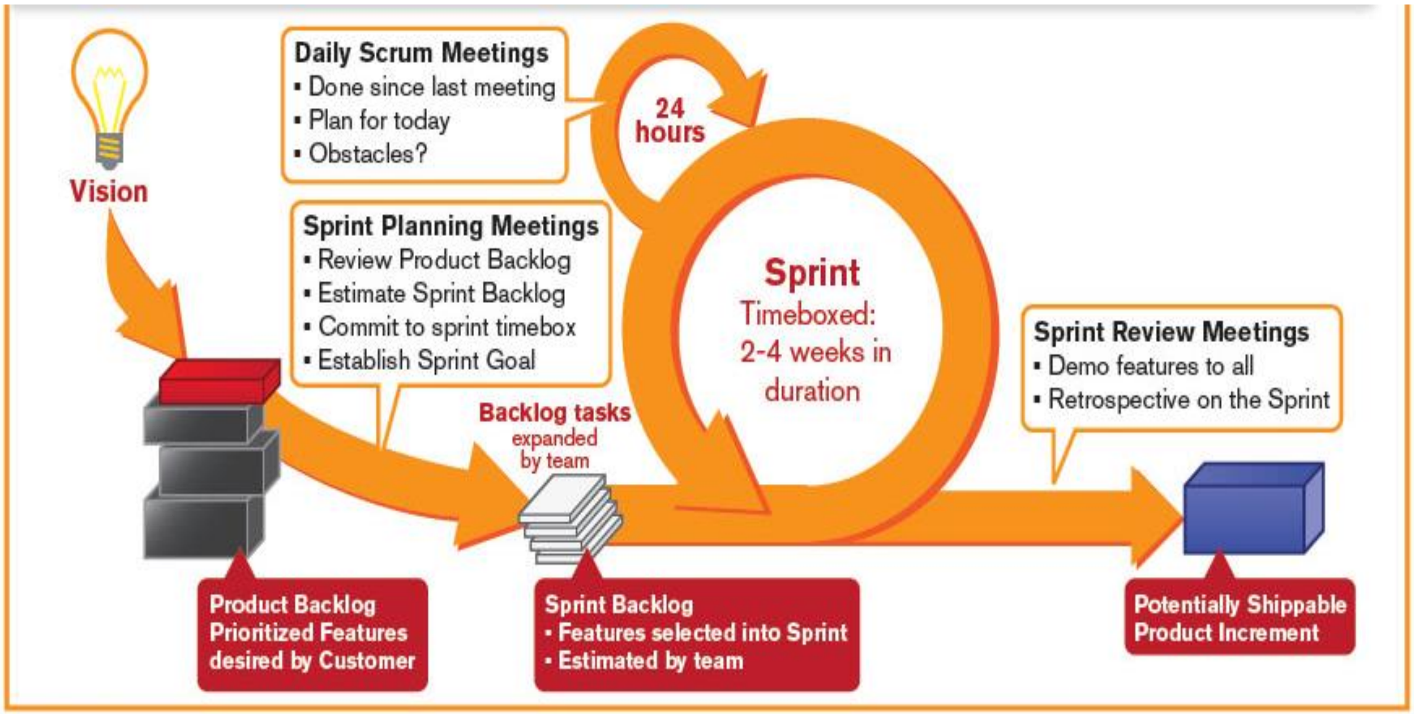
## Agile……Cont.

- A **product owner** creates a prioritized wish list called a **product backlog**.
- During **sprint planning**, the **team** pulls a small chunk from the top of that wishlist, a **sprint backlog**, and decides how to implement those pieces.
- The team has a certain amount of time, a **sprint**, to complete its work - usually two to four weeks - but meets each day to assess its progress (**daily scrum**).
- Along the way, the **ScrumMaster** keeps the team focused on its goal.
- At the end of the sprint, the work should be **potentially shippable**, as in ready to hand to a customer, put on a store shelf, or show to a stakeholder.
- The sprint ends with a **sprint review** and **retrospective**.

As the next sprint begins, the team chooses another chunk of the

# Agile – A Sprint View

- Scrum is an iterative and incremental process so the project is split into a series of sprints
- Scrum is ideally used for projects with rapidly changing or evolving requirements in SW projects
- With Scrum, projects progress through iterations called sprints. Each sprint can be 2-4 weeks long
- Sprint backlog is the team’s to-do list for the sprint


Vision
Daily Scrum Meetings
▪ Done since last meeting
▪ Plan for today
▪ Obstacles?
24 hours
Sprint Planning Meetings
▪ Review Product Backlog
▪ Estimate Sprint Backlog
▪ Commit to sprint timebox
▪ Establish Sprint Goal
Sprint
Timeboxed:
2-4 weeks in duration
Sprint Review Meetings
▪ Demo features to all
▪ Retrospective on the Sprint
Backlog tasks expanded by team
Product Backlog Prioritized Features desired by Customer
Sprint Backlog
▪ Features selected into Sprint
▪ Estimated by team
Potentially Shippable Product Increment

## Requirement Engineering

- Descipline of gathering, analyzing and formally specifying the user's needs. Must be oriented towards the user's needs
- Small change to requirement can profoundly affect the project's: Cost; Schedule; Implementation strategy
- Requirements develop over time
- Important to ensure they are not frozen in the beginning of a project
- Requirements should be well understood as far as testability is concerned
- Requirement traceability or validation matrix is important

## Requirement Engineering… cont.

- Users know their domain, involve them from the beginning!
  - ✓ Define functions; System design; Implementation; User Testing
- User should participate in creating, verifying and updating the requirements
- Project manager’s job is to ensure the relationship with the users is maintained
- It is important the requirements reflect user’s work and the tasks the SW system under development is supposed to automate

## Requirement Engineering… cont.

### Gathering of User Requirements

- Inquiry based requirement gathering: Structured interviews and questionnaires that user fills
- Diagram based requirement analysis: using diagrams to sketch user's work process and describe requirements graphically
- Scenario based: a typical sequence of activities that outlines what the user will do with the system
- Prototype system development: to make requirements clear and establish better understanding

## Requirement Engineering… cont.

### Gathering of User Requirements… cont.

- **Analysis through videotaped user’s work process**
- **Gathering requirements is product related activities. The validation activities, when carried out, assists you in ensuring you have built the product according to the requirement and as stated in the requirement document**

## Requirement Engineering…cont.

**Processes captured through use-cases and use diagrams**

- **Focus on user's interaction with the product or technology**
- **Textual description of how a user might use a function of the system. It includes system responses and user actions**
- **The name of the Use Case should be short and descriptive, and should be stated in Verb-Subject order. For example, *Pay Invoices* is preferable to *Invoice Payment***

## Requirement Engineering…cont.

Processes captured through use-cases and use diagrams

- *Pay Invoices* emphasizes that something is being done rather than just being some functionality, decomposed set of tasks

- Use Case should describe what the actor does and what the system does in response (Use Case is phrased in the form of a dialog between the actor and the system)

## Software Architecture

- SW architecture consists of specification and design of system
  - ✓ Components: modules, objects, files
  - ✓ Connectors: define interactions amongst components through procedure calls, parameters of initialization etc

- Architectural choices:
  - ✓ Event-based: chronological sequence of events on a system
  - ✓ Component-based: Code, in source or executable form, with well-defined interfaces and behavior
  - ✓ Real-time
  - ✓ Object-oriented
  - ✓ Pipeline architecture: a set of data processing elements connected in series, so that the output of one element is the input of the next one

## Software Architecture… cont.

- Important for system architecture to support all use-cases
- System architecture should take the following things into consideration:
  - Platform the system is going to run on
  - Hardware architecture
  - Operating system
  - Database management system
  - Network protocols
  - Data communications
  - Legacy systems
  - System’s functionality, reliability and usability

## Staffing

- Staffing of development teams and assigning roles
- Staff training
- Principle of independence: open to innovation and change
- New Technologies
- Collaboration: intensive teamwork
- Alignment: each individual in an organization understands common vision and supports to adhere the processes for SW development
- Professional culture: Open sharing and mutual respect

## Management Strategies & Techniques

- Balance the need for process control with need of flexibility
- Establish SW measurement programs
- Make broader strategy of a company and management objectives clear to the development teams
- Identify most critical issues of the project and allocate additional development resources, time and efforts to them
- Establish formal management process for reviewing and approving product enhancements: CCB

## Management Strategies & Techniques… cont.

- Promote orientation towards strategic partnership
- Establish project review mechanisms
- Insist on creating multiple design reviews: functional, object-oriented, event-based, and data-flow
- Multiple design reviews help uncover different perspectives

## Session: 5
## Prototype

- **Key activities:**
  - **Determine the requirements**
  - **Develop early model of prototype with user interfaces**
  - **Conduct review with the end-users**
  - **Based on the feedback, determine the scope of the project, contractual obligations and proceed**

## Scope of Prototype

- Horizontal Prototype
  - Provides general view of the whole system, focusing on user interactions
  - Gives opportunity to fine-tune the system based on user feedback
  - Helps develop cost and schedule estimates

- Vertical Prototype
  - A view of a single feature and provides opportunity to finalize details of the functions, eg. System interface requirements, data loads, user volume
  - Provides opportunity to understand finite details of a specific feature

## Types of Prototypes

- Evolutionary Prototype: Development team can continually make changes which were not thought of in previous phases of design or requirements
- Visual Prototype: These are in the form of screen mock-ups, perhaps in a paper form or created using a graphics tool
- Incremental Prototype: The product is built as separate prototypes and at the end merged into one
- Throwaway Prototype: The model does not become part of the final system. It is used to gather requirements and show the users a model of the finished system

## Risk Assessment

- Estimation on project size
- Setting up the deadlines realistically
- Collecting and studying reports on other similar projects
- Top management commitment: if they don't take interest in the project, all other risks will be impossible to address
- Management change
- Failure to gain user commitments
- Requirement change and timeliness of the change: this can turn into a costly proposition!

## Risk Assessment… cont.

- Familiarity with technology: the higher the experience with application languages, technology databases, hardware and operating systems, the lower the risk to the project
- Insufficient/inappropriate staffing
- Staff turnover
- Hardware unavailability
- Size underestimate

## Risk Management

Risk management addresses risk types such as project related, product-specific or business risks

- Project related risks can be related to funding, team member availability, sufficient testing resources
- Product risks affect the quality or performance of the software, eg. the quality of program code or changes in requirements
- Business risks are related to the changes in economic conditions or management decisions

## Risk Management… cont: Risk Analysis

- **Risk analysis involves determining and assessing probability and impact**
- **Usually probability is calculated with statistical numbers**
- **Calculated risk ranks in the order engineers believe it will occur**
- **The "weighting" of a risk can be done based on the impact as negligible, marginal, critical, or catastrophic**

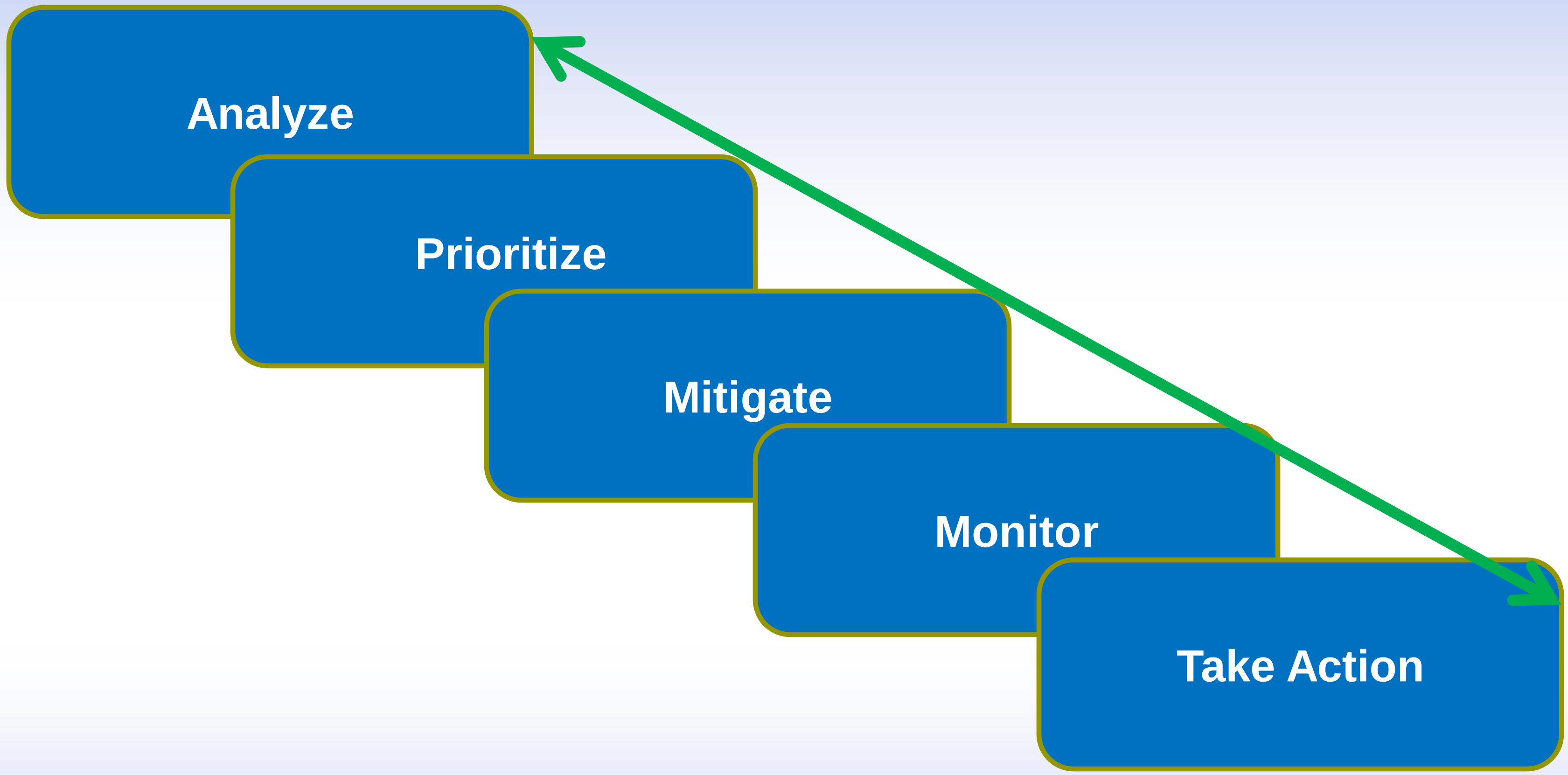
## Risk Management… cont: Prioritization

- Review the probability of the risk occurring
- Next, determine the cost to the project in dollars if the risk occurs
- Then assign a cost to address the risk
- For example, losing a senior engineer in the middle of a project can affect the quality of program code and delay release of the software program and cost
- Mitigation: If the project has additional, qualified team members to work on it, dividing the workload between other team members will help reduce the risk

## Risk Management… cont.

- The pre-requisites and dependencies of Work packages and tasks determine a critical path
- It is the sequence of activities that takes the longest time to complete
- Any delay to an activity in the critical path will cause delays to overall project
- Delays to activities not on the critical path may not cause overall delays

## Risk Management Roadmap



## Software Metrics

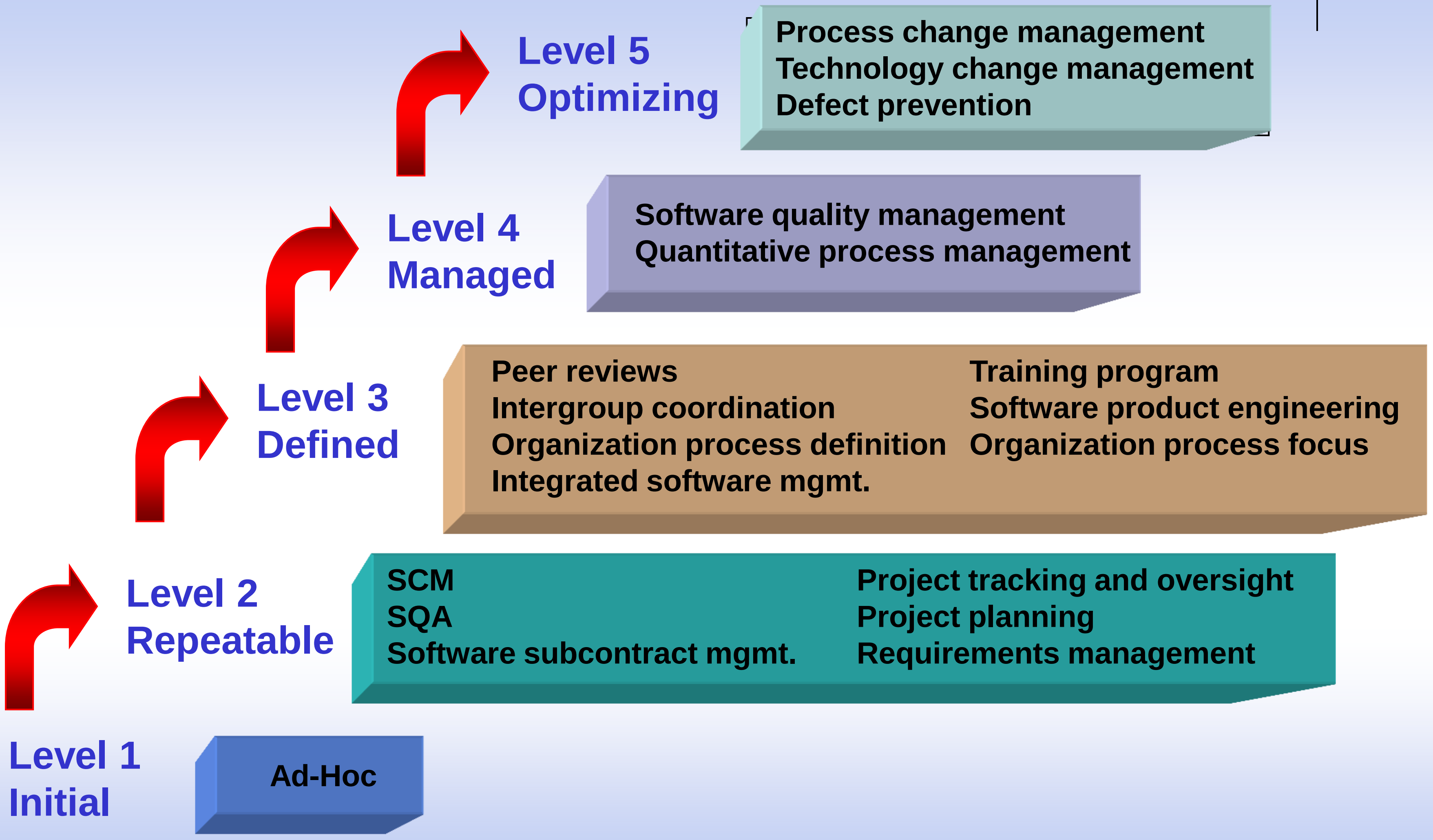
**Metrics for:**

- Project size and team productivity
- Schedules
- Requirement specification
- Software testing: number of defects
- Software quality: phase escape
- Project risks: complexity

## Standards

- External Standards such as ISO
- ISO 9000 series of Standards provides generic model for Quality Management Systems
- SEI's CMMI: Used to establish a maturity level of an organization's software processes
- Organization's own lifecycle processes

# Decision on implementing SEI Maturity Model



**Level 5 Optimizing**
- Process change management
- Technology change management
- Defect prevention

**Level 4 Managed**
- Software quality management
- Quantitative process management

**Level 3 Defined**
- Peer reviews
- Intergroup coordination
- Organization process definition
- Integrated software mgmt.
- Training program
- Software product engineering
- Organization process focus

**Level 2 Repeatable**
- SCM
- SQA
- Software subcontract mgmt.
- Project tracking and oversight
- Project planning
- Requirements management

**Level 1 Initial**
- Ad-Hoc

## Additional Information

- Productivity Measurement & Estimate Validation, IBM Corporation, Allan Albrecht
- Controlling SW Projects, Yourdon Press, Tom DeMarco
- Function Point Analysis, Prentice Hall,  Brian Dreger
- Practical SW Metrics for Project Management and Process Improvement, Robert Grady, Prentice Hall
- SPQR/20 Users Guide, SW Productivity Research, Capers Jones
- Rethinking the SW Process, CD-ROM, Miller Freeman, Lawrence Kans
- SW Benchmark Studies, 1997, Howard Rubin and Associates
- Best Practices in SW Costs and Schedule Estimation, Prentice Hall, William Roetzheim
- Software cost estimation with Cocomo II, Prentice Hall, Barry Boehm

The End!!