



Development Process

Mikael Svahnberg¹

2016-03-09

¹Mikael.Svahnberg@bth.se



Software Engineering

- IEEE std 610.12:1990 “IEEE Standard Glossary of Software Engineering Terminology”:

Software Engineering

The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software.



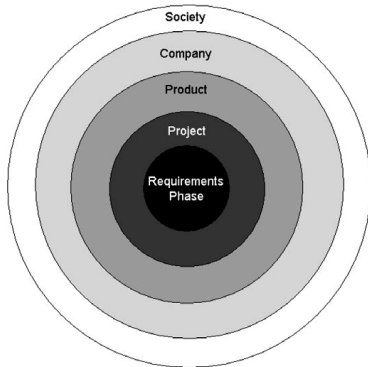
Software Engineering Process

- **Systematic**
 - Pre-planned, not ad-hoc
 - Thorough
 - Repeatable
- **Disciplined**
 - Following the plan
 - Eyes on target
- **Quantifiable**
 - Measurable
- **Development**
 - *this
- **Operation**
 - Deployment is an important part of SE, and must be planned accordingly.
- **Maintenance**
 - 80% – 90% of a system's life span is spent in maintenance.



Process vs Project vs Product

T. Gorschek, A.M. Davis, *Requirements Engineering; In Search of the Dependent Variables*, Information and Software Technology 50(2008):67–75.



(+ Process, which is not visible in this figure but neatly bisects it.)



Example of UML Process:

Dice Game Machine

- On the Machine a player may login, logout or play the game.
- When playing the game a player rolls two die. If the total number of points is greater than seven the player wins, otherwise the player loses.

Construct

- Use Case Diagrams
- Use Cases
- Conceptual Model
- Class Diagram
- Collaboration Diagram
- Interaction Diagram
- Flowcharts?
- ?? What happened to testing ??



Discussion

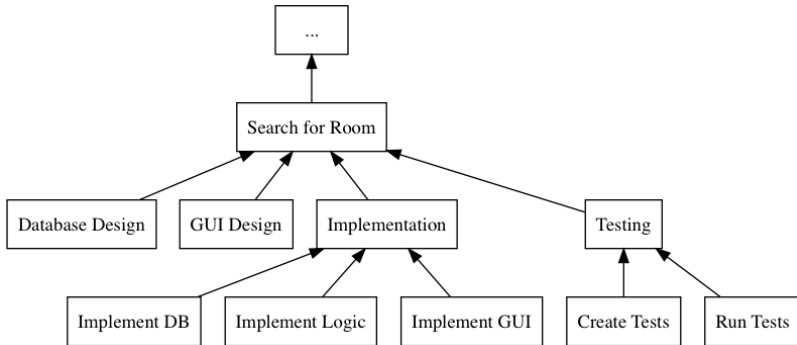
- What is good with waterfall?
- Where/How would you do design in Scrum?
- Where would you do design in Kanban?
- When should you use which process model?
- What are their limitations?
- Does it work to incrementally test a product like this?



Project Planning

- What do we need to know in order to plan something?
- How do we put this together into a plan?

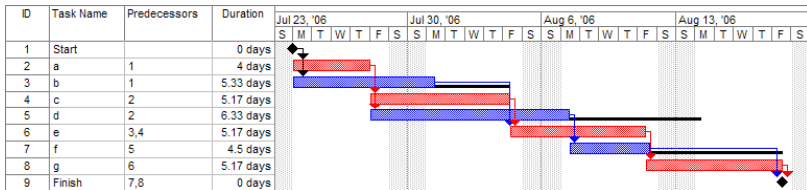
Work Breakdown Structure





GANTT

Feature	Tasks	Sub-Tasks	Effort	Start Date	End Date	Responsible	Spent Time	Progress	Projected Effort	Over/Undertime
Search for Room	Database Design								spent/progress	(est. eff.) - (proj. eff.)
	GUI Design									
	Implementation	Implement DB Implement Logic								
	Testing	Implement GUI Create Tests Run Tests								





Tracking Progress

- Reporting *Time* or reporting *Progress*
 - Amount of time/money spent
 - Delivered LOC?
 - Completed Tasks?
- Earned Value Charts
 - Planned cost (value)
 - Actual cost
 - Earned Value

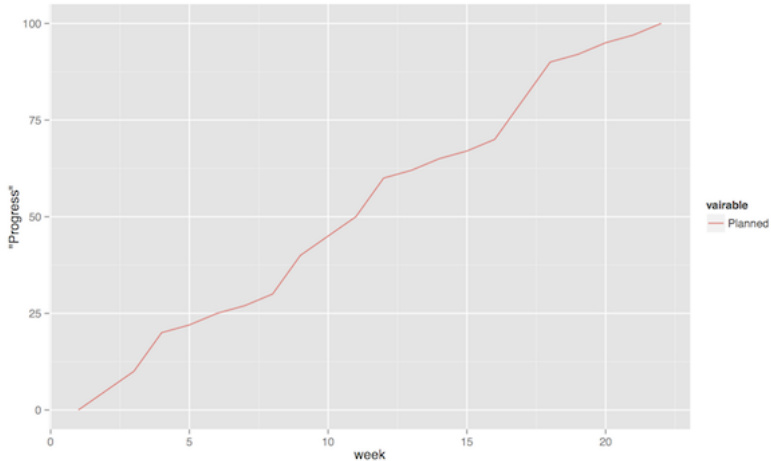


Story Points

- An arbitrary measure of the size of a task
- Typically uses a modification of a fibonacci sequence:
 - 1, 2, 3, 5, 13, 40, 100
- Use them to
 - measure *velocity* of your development team.
 - plan sprints accordingly

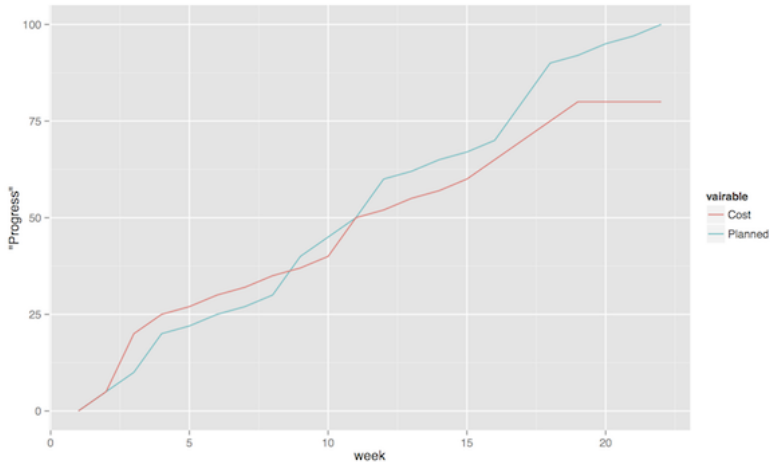


Earned Value Charts: Planned



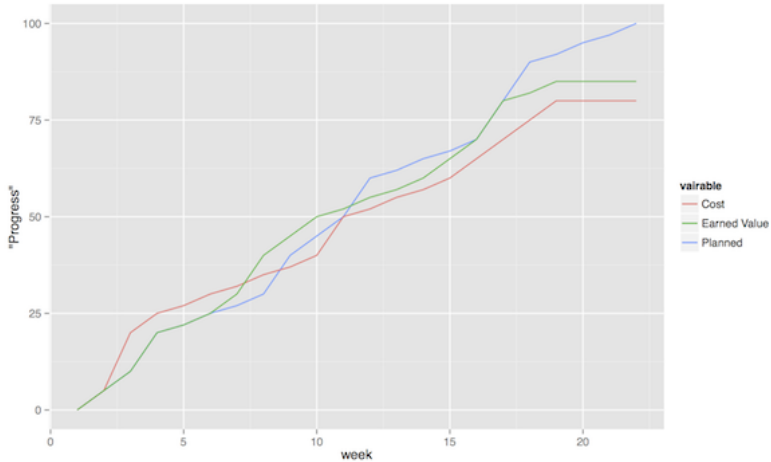


Earned Value Chart: Adding Actual Cost



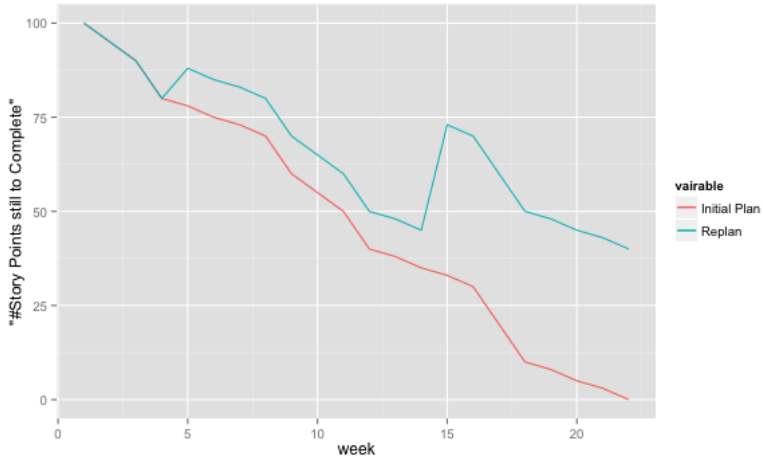


Earned Value Chart





Burndown chart





Risk Management

- Identify risks
- Develop plans to minimise their effect on a project
- A risk is a probability that some adverse circumstance will occur
 - Project risks affect schedule or resources
 - Product risks affect the quality or performance of the software being developed
 - Business risks affect the organisation developing or procuring the software
- Monitor and mitigate risks