Software Project Management
CS6704: Class 2

Instructor: Shawn A. Bohner
Voice: (703) 538-8374
Email: sbohner@vt.edu

Teaching Assistant: Yunxian Zhou
Email: yxzhou@nvc.cs.vt.edu
Voice: (703) 538-8381

Agenda

- Review Last Week’s Material
  - Turn in Homework
  - Reading Discussion
  - Project Plans Discussion
  - Review last weeks class
- Chapter 1: What Makes a Good Software Manager?
- Chapter 2: Four Basics that Work
  - Break
- Software Process Management
- Homework Assignment
Fall Semester Timeline

Aug  —  Sept  —  Oct  —  Nov  —  DEC

Class Begins
PM Basics
Managing with Metrics
Emerging PM Paradigms
Software Project Planning
Mid-Term Exam
Program Management
Final Exam
Software Process and Life Cycle
Software Estimation
Risk Management
Human Side of PM
Project Portfolio Management

16 weeks, 13 sessions... So much to do & so little time... manage your time effectively!

Discussions...

◆ Reading Discussion
  ● What was the main idea of the paper?
  ● Does software projects differ from other projects? If so, how?
  ● Do you do these things in your projects?
  ● What parts of these can be automated?

◆ Software Project Plans Discussion
  ● Where did you find software plan examples?
  ● What did the software plans contain?
  ● Was it what you expected? What was different or missing?
  ● What there enough information in the plans to support the management decisions for projects you have seen?
So, why is e-Engineering a project management problem?

- Large, complex, distributed systems
  - CPU speed, memory, bandwidth
  - Things we’re taught to ignore
- Heterogeneous platform architectures
  - Mainframe, desktop, laptop, palmtop
  - CORBA, VXML, EJB, ... languages
- Mobility (Several 2000/1 IEEE Computer Issues)
  - One application runs on many devices across varied communications
  - Each device has varied resource profiles
- Modeling, analysis, simulation...
  - Collaborative design on partial information
- Development teams across the globe
  - Around the clock development
  - Decentralized ownership and control

Why Are Projects Late?

- Unrealistic deadline established by someone outside the software development group
- Changing customer requirements that are not reflected in schedule changes
- Honest underestimate of the amount of effort and/or the number of resources that will be required to do the job
- Predictable and/or unpredictable risks that were not considered when the project started
- Technical difficulties that could not have been foreseen in advance
- Human difficulties that could not have been foreseen in advance
- Miscommunication among project staff that results in delays
- Failure by project management to recognize that the project is falling behind schedule and a lack of action to correct the problem

Which of these are unique to Software Projects?
So, what did Capers Jones tell us here?

<table>
<thead>
<tr>
<th></th>
<th>Early</th>
<th>On-Time</th>
<th>Delayed</th>
<th>Canceled</th>
<th>Sum</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 FP</td>
<td>14.68%</td>
<td>83.16%</td>
<td>1.92%</td>
<td>0.25%</td>
<td>100.00%</td>
</tr>
<tr>
<td>10FP</td>
<td>11.08%</td>
<td>81.25%</td>
<td>5.67%</td>
<td>2.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>100FP</td>
<td>6.06%</td>
<td>74.77%</td>
<td>11.83%</td>
<td>7.33%</td>
<td>100.00%</td>
</tr>
<tr>
<td>1000FP</td>
<td>1.24%</td>
<td>60.76%</td>
<td>17.67%</td>
<td>20.33%</td>
<td>100.00%</td>
</tr>
<tr>
<td>10,000FP</td>
<td>0.14%</td>
<td>28.03%</td>
<td>23.83%</td>
<td>48.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>100,000FP</td>
<td>0.00%</td>
<td>13.67%</td>
<td>21.33%</td>
<td>65.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>Average</td>
<td>5.53%</td>
<td>56.94%</td>
<td>13.71%</td>
<td>23.82%</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

From Capers Jones, *Patterns of Software Systems Failure and Success*  

Semi-Formal Definition

- **Management** – The activities undertaken by ??? to plan and control activities of others to ???.

- **Project Management** – “… a ??? of procedures, practices, technologies, and know-how that provide the ?, ??, ???, ?????, and ???? necessary to successfully manage an engineering project.” [Thayer 1987]
Classic Management Activities

- Planning
- Directing
- Staffing
- Organizing
- Controlling

Software Project

Prog. Mngt. Capability Maturity Model

<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Key Process Area Concentrations</th>
<th>Strategic Inflection Point</th>
<th>Effective Span</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Incorporated</td>
<td>Value Management, Business Continuity Planning, Procurement Management, Outsourcing and Contract Management, PM Center of Excellence</td>
<td>Integration with Business</td>
<td>Enterprise/Industry</td>
</tr>
<tr>
<td>4 Managed</td>
<td>Program Process Management, Project Integration Management, Project Performance Management, Vendor Management, PM Career Path, Staff Performance Management, Customer Relationship Management, Contingency Management, Communications Management</td>
<td>Dynamic Micro-Level Change</td>
<td>Multiple Business Units</td>
</tr>
<tr>
<td>3 Defined</td>
<td>PM Methodology, Skill Management, PM Training, Risk Management, Change Management, Staff Resource Management, Environment Resource Management, Conflict/Issue Management</td>
<td>Static Macro-Level Change</td>
<td>Multiple Project</td>
</tr>
<tr>
<td>1 Initial</td>
<td>Acquiring New PMs</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Result

Value

Risk
What do PMs Manage? Software!

- Software is part of a computer system that is ???
- Business changes and its computer systems must respond
- Software is ???, not manufactured
- Software is intangible
- Software is complex

Software is suppose to change... otherwise it would in the hardware!

Software Project Management must Manage to a Future (moving) Target

- ??? Changes
  - Market shifts and investment fluctuations
  - Portfolio changes
  - Mergers and Acquisitions ...
- ??? Changes
  - Hardware, software, networks, mobile ...
  - Fast, better, cheaper ...
- ??? Changes
  - Agility vs. quality
  - Business and technology ...
- ??? Changes
  - Shortages and gluts ...
Project Management Truths

- You can get someone to commit to an unreasonable deadline, but you can’t
  - ...bully him into meeting it
- Experienced PM’s Motto:
  Projects fail
  - ...one day at a time
- Projects progress to 90% completion very rapidly...
  - They remain there while great sums of money and time are spent to keep them there
- You can freeze specifications,
  - ...but you can’t freeze expectations
- When quoting “off the shelf,” remember
  - ...there is no shelf
- If project content is allowed to change freely,
  - the rate of change will soon exceed the rate of progress

Chapter 1: What makes a good Software Manager?

- The purpose of this chapter is to examine characteristics of a good software project manager
  - What are the key attributes associated with successful project managers?
  - Which software projects management skills are essential?
- Objectives
  - Outline key characteristics of a good software manager
  - Examine people, business, and process perspectives
People Perspective

- Rob Thomassett – “most projects fail because of people and PM concern rather than technical issues.”
- ~80% of software project managers come from technical ranks
  - Some with natural abilities, others must learn
  - Often with predispositions about managers and how projects should be run
  - Most are surprised by their first project
    - Budgets, resources, customers, estimates, teams, meetings, decisions, and risks
- Beware of the little hairy boss syndrome...

Some Sage Advice...

- Be flexible.
  - Let your people perform according to their capabilities. Stretch goals are good but they have to have enough room to stretch.
- Have compassion.
  - Deal compassionately with difficult people (often they do not know they are difficult and may be difficult because the do not understand)
- Know when to lead and when to manage.
  - Lead people... manage process and product (by example) systematically
- Accept the role of meetings.
  - Communication, not bureaucracy
  - Prepare and manage them
Coach Your Team to Success [Hawkins 1994]

- Yes, project management is a team sport
- Success is spelled “T E A M”
- Increase chances of success by
  - Hiring the best and brightest
  - Keep the teams small
  - Minimize distractions
  - Train staff for best performance
  - Meet as a team often but not for long periods
  - Know your team members
  - Set the example

Software Teams

Consider the following factors when selecting a software project team structure ...

- Difficulty of the problem to be solved
- Size of the resultant program(s) in lines of code or function points
- Time that the team will stay together
- Degree to which the problem can be modularized
- Required quality and reliability of the system to be built
- Rigidity of the delivery date
- Degree of sociability (communication) required for the project – Relationship management
Business Perspective

- Again, most software project managers did not come from the business...
- Chaos Study
- Creating software products to benefit the business and its customers
  - Who wants the software and who doesn’t?
  - Who are the users? What will the software do for them and the business?
  - When do users need (not want) the software?
  - Why does the business need the software?
  - Why is your team developing the software?
- Business investment portfolio view

Basis for IT Portfolio Investment

- Value maintenance — managing ongoing, non-discretionary investments in IT assets
- Value enhancement — discretionary investments in improving or growing IT asset base
- Value exploration — venture into high-risk/high-payoff IT investments
Maintain Existing IT Asset Value

- IT liability avoidance and value retention
- Fund baseline costs for critical business operations, maintenance, and support
- Skeleton funding based on minimum headcount and costs to keep system running
- Incentives to reduce baseline costs

Enhancing Existing IT Asset Value

- Strategic priorities
  - Investments criteria
  - Investment process
- Phased funding on projects with interim deliverables
  - Jump-start
  - Initial capability
  - Advanced function
- Continued allocations through project value and delivery commitments being met and communicated
Exploring Future IT Asset Value

- Requires dynamic mindset for quick response to value and market changes
  - Digital Planning for agility
- Value Enhancement investment criteria plus:
  - Business/Market advantage
  - 2-3 year ROI/IRR projection
  - Clear exit strategy
- Manage using a venture funding model – close and regular interactions

Engage in high risk, high yield opportunities

Value Exploration
Value Enhancement
Value Maintenance

Investment Model IT Portfolio

Excess Investment
Lost Value
Value Maintenance
Value Enhancement
Leveraged Value

Cost to Conform
Followers

Cost to Excel
Leaders
Process Perspective

- Software process has matured into a vital part of software projects
  - Software Engineering Institute’s Capability Maturity Models
  - Best practices
  - ISO 9000, Malcolm-Baldridge, TQM...
- At the heart of every project plan is an activity/task set derived largely from the process

Management Secrets

- Avoid having team members work in isolation
- Stay with your project team – they are the ones delivering the products of the project
- Concentrate on Tasks – not tools
- Do your homework (no this is not a subliminal message for CS6704)
  - Stay current on latest advancements in management and technical techniques for projects that you manage
- Sounds like common sense… but it is not so common!
Chapter 2: Four Basics that Work

◆ The purpose of this chapter is to examine four basic software project management principles that work
  ● What are the basic areas for successful project managers to manage?

◆ Objectives
  ● 1. Balance people, process, and product
  ● 2. Promote visibility
  ● 3. Organize using configuration management
  ● 4. Apply standards judiciously

The 3 P’s

◆ People
  ● Critical to all projects but the most variable in the management equation
  ● Creative, complex, capable, costly

◆ Process
  ● Most focused on in recent years
  ● Manage the process through measures (next section)
  ● Universe, world, and atomic views

◆ Product
  ● Software is to be change tolerant
  ● Quality measured
  ● Ultimate delivery
Balancing the 3 P’s

- Difficult of the product dependent on people and process capabilities
  - Must triangulate on Pfactors...
- People Knowledge
  - Domain
  - System
  - Programming
- Process Maturity
  - Levels 1-5 on the SEI-CMM scale
- Product Maturity
  - From research prototype to packaged component

Visibility

- Software is invisible to naked eye...
  - Intangible – must be measured with a computer
- Software projects are largely invisible (until they complete) – managed as a cost
- Project managers must bring visibility to software product and project
- Two “dreaded” visibility vehicles
  - Documentation
  - Metrics
- Project team vision, commitment, and group memory
Visibility Through Modeling

- Models answer questions
- What questions need clarity (better visibility) – model them
- Modeling tools
  - SADT/IDEF0, Warnier-Orr diagrams, STDs, Structure charts
  - CRC cards, object interaction diagrams
  - Mind maps, story boards,

How are Metrics Used for Visibility?

- Solving problems — Which choice or improvement should be made?
  - Benchmarking for performance improvement
- Getting attention — What situations need to be addressed?
  - Dashboard of indicators
- Keeping score — How well is it (or IT) doing?
  - Scorecard on goals
Meaningful Management Visibility

**Executive Decision View**
- ROI, ROM, EVA, ...
- Business Impact
- Price/performance
- Risk/opportunity ... Value

**Management View**
- Costs/budget
- Schedule/effort/delay
- Utilization and loading
- Resource availability ...

**Operational View**
- Process/activities
- Products/specs
- Policy/procedures
- Constraints/guides ...

Key Visibility Principles for Metrics
- Clearly defined metrics, consistently applied
- Metrics are only indicators, use them accordingly
- Focus on leading indicators over lagging ones
- Recognize indicators of problems
  - Lack of change
  - Frequent change
  - Slow, steady deviation from plans

*Software metrics are navigational instruments giving position, direction, and rate of change*
Configuration Management

No matter where you are in the system life cycle, the system will change, and the desire to change it will persist throughout the life cycle.

*Ed Bersoff, et al, 1980*

- Product dependencies can be complex
  - Time, use, version, …

---

Flow of SCM: Definition, Use, Archive

<table>
<thead>
<tr>
<th>SCM Staff</th>
<th>Software Project Staff</th>
<th>SCM Staff</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defines SCM Structures, Standards and Procedures</td>
<td>Uses SCM Library Structures, Standards and Procedures</td>
<td>Archives/Controls Baseline Software</td>
</tr>
</tbody>
</table>

Diagram showing the flow of SCM with libraries for Production, Acceptance Test, Integration Testing, System Testing, Unit Development, and Emergency Fix libraries managed by SCM Staff and Software Project Staff.
Standards

- The wonderful thing about standards is that there are so many to choose from...
  - But when leveraged correctly, they simplify the process and management of a software project
- Standards save time and money
- Examples:
  - IEEE 1042 – SCM Standard

Software Process

- The purpose of this module is to introduce several software process models used to manage large-scale software projects
  - While no software process works well for every project, every project needs to conform to some systematic process
- Objectives
  - Outline software as a process
  - Outline key aspects of software process
  - Examine how to assess software process maturity
The Linear Model

System/information engineering

analysis  design  code  test

Iterative Models

listen to customer

build/revise mock-up

customer test-drives mock-up

Prototyping

60 - 90 days

RAD

testing & turnover
The Incremental Model

An Evolutionary (Spiral) Model
Still Other Process Models

- Component assembly model—the process to apply when reuse is a development objective
- Concurrent process model—recognizes that different parts of the project will be at different places in the process
- Formal methods—the process to apply when a mathematical specification is to be developed
- Cleanroom software engineering—emphasizes error detection before testing

SEI Process Program Background

- CBA IPIs and SPAs conducted since 1987 through December 1999 and returned to the SEI by January 2000
  - 1512 assessments
    - 1024 CBA IPIs
    - 488 SPAs
  - 1166 organizations
  - 309 participating companies
  - 283 reassessed organizations
  - 6168 projects

*Source: Software Engineering Institute*
**SEI’s Software Process Capability Maturity Model**

<table>
<thead>
<tr>
<th>Level</th>
<th>Focus</th>
<th>Key Process Areas</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Optimizing</td>
<td>Continuous Process Improvement</td>
<td>Defect prevention, Technology innovation, Process change management</td>
<td>Productivity &amp; Quality</td>
</tr>
<tr>
<td>4 Managed</td>
<td>Product and Process Quality</td>
<td>Process measurement and analysis, Quality management</td>
<td></td>
</tr>
<tr>
<td>3 Defined</td>
<td>Engineering Process</td>
<td>Organization process focus, Organization process definition, Peer reviews, Training program, Intergroup coordination, Software product engineering, Integrated software management</td>
<td></td>
</tr>
<tr>
<td>2 Repeatable</td>
<td>Project Management</td>
<td>Software project planning, Software project tracking, Software subcontract mgmt, Software quality assurance, Software configuration mgmt, Requirements management</td>
<td></td>
</tr>
<tr>
<td>1 Initial</td>
<td>Heroes</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Source: Software Engineering Institute*

---

**Summary of the SEI/CMM Levels**

1) **Initial**: The software process is characterized as ad hoc, and occasionally even chaotic. Few processes are defined, and success depends on individual effort.

2) **Repeatable**: Basic project management processes are established to track cost, schedule, and functionality. The necessary process discipline is in place to repeat earlier successes on projects with similar applications.

3) **Defined**: The software process for both management and engineering activities is documented, standardized, and integrated into a standard software process for the organization. All projects use an approved, tailored version of the organization’s standard software process for developing and maintaining software.

4) **Managed**: Detailed measures of the software process and product quality are collected. Both the software process and products are quantitatively understood and controlled.

5) **Optimizing**: Continuous process improvement is enabled by quantitative feedback from the process and from piloting innovative ideas and technologies.
Summary Maturity Findings:
Organizational Trends

- **Software Quality Assurance** is the least frequently satisfied level 2 KPAs among organizations assessed at level 1
- **Integrated Software Management, Training Program and Organization Process Definition** are the least frequently satisfied level 3 KPAs among organizations assessed at level 2
- **Higher maturity has been reached** among those organizations reporting reassessments

*Source: Software Engineering Institute*
Process Improvement Maturity Levels

**Level 1: Ad hoc**
- High variability/Low predictability
- Heroes => Success
- High Risk

**Level 2: Repeatable**
- Project metrics focus
- Low variability/Medium predictability
- PM => Success
- Medium - High Risk

**Level 3: Defined**
- Process metrics focus
- High predictability
- Process => Success
- Medium Risk

**Level 4: Managed**
- Product and Process metrics
- High predictability
- Managed Process => Success
- Low-Medium Risk

**Level 5: Optimized**
- Value metrics
- High predictability
- Agility => Success
- Low Risk

More Traction at Upper levels...
Homework Assignment for 9/10/01

- Read Paper entitled, “Anchoring the Software Process” by Barry Boehm
  - Summarize key points of assigned reading into a short 1 page paper (~300-500 words)
- Read SPMH Text Chapters 1, 2, and 3
  - Homework Questions/Deliverables:
    1. What are the key characteristics of an effective project manager? Please include references.
    2. Outline how to balance people, process, and product for effective project management
    3. What visibility techniques work best for showing project progress? Why?
    4. Search web for Configuration Management Plan Template and send it to me via email.

- Have a great week!