01: Introduction

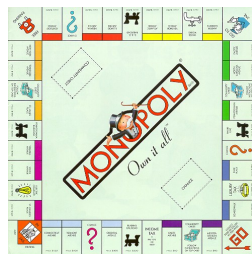Software Project Management
Philippe Kruchten

1

Module outline

- What is a project
- What is Project Management
- What is a software project
- What is software project management
- A brief history of project management
- The special case of *software* projects

2

- Software

- Project

- Management

3

What is a project?

- An endeavor with a defined goal, a start and an end, and some constraints.

4

What is a Software Project

- Software-intensive system
- Software-reliant system

- Software represents the dominant cost

- Focus here is on the *software* part, though many aspects will apply to the whole system

5

Software Project

- Objectives
  – Clearly defined set of goals; non conflicting…?
- Start and end points
  – not a continuous activity, like operations or support
- Uniqueness
  – One time thing (product); not a repetitive activity
  – Not production, operations
- Constraints
  – Cost, schedule, quality

Source: Futrell

6

What is Project Management

- Project management is the application of knowledge, skills, tools, and techniques to project activities in order to meet or exceed stakeholder needs and expectations from a project
(Project Management Institute)

Defining Software Project Management

- Software Project Management is the art of balancing competing objectives, managing risks, and overcoming constraints to successfully deliver a product which meets the needs of both customers (the payers of bills) and the users. (RUP)

"Clear end point"…?

- Multiple releases
- Continuous deployment

- Resources (financial, staff)
- Difference between:
    – Internal IT support
    – Software product (or software in a system)
    – "Bespoke" software (contract with a customer)

Origins of project management

- Pyramids, Roman bridges, and all that stuff.

- Henry Gantt, 1917: barcharts
- Flow line scheduling: 1930
- Line of Balance: 1940 Goodyear, US Navy
- Milestone charts, 1940
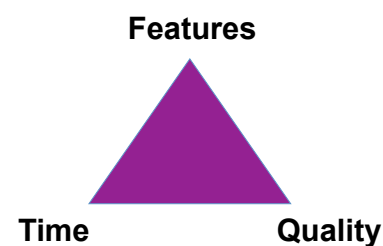- Critical Path Method (CPM), DuPont de Nemours 1950's
- PERT, US Navy 1958

Origins

- Precedence method
- Iron triangle (time, cost, output):
    – *Martin Barnes, UK, 1969*
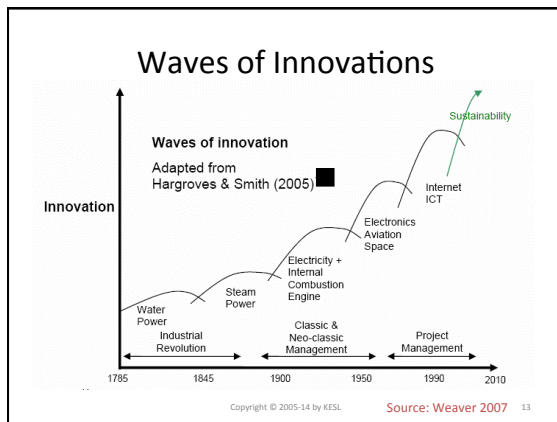- …
- …
- Toyota Production System
- Agile Manifesto

A Fundamental Conundrum

Features

Time   **Quality**

Waves of Innovations

Source: Weaver 2007   13
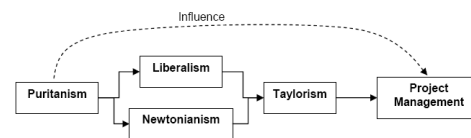
Pre-classical developments

- Puritanism: the protestant work ethic
- Liberalism: capitalism, division of labour
- Newtonianism: scientific enquiry

Source: Weaver 2007

Copyright © 2005-14 by KESL   14

Classical development

- Taylorism: scientific management
  – Frederick Taylor (1911)
  – Work breakdown structure and scheduling
- Bureaucracy, administrative management
- System theory, contingency view (1940's)
  – Multiple input, multiple outputs, complex interrelated processes
- Operations research
  – *Military tactics, supply, etc…*

Copyright © 2005-14 by KESL   15

And then came the computer

- Mainframes

- PC (commodore, Atari, Apple, IBM)
- Microplanner
- Primavera (1983)
- Microsoft Project (1987)

- *And now everybody can schedule… yeah!*

Copyright © 2005-14 by KESL   16

And then….

- Earned Value Management
- Program management
- Portfolio management

Copyright © 2005-14 by KESL   17

On the bleeding edge…

- Lean production
  – And therefore "lean management"
- Critical Chain and Theory of Constraints
- Chaos theory
  – Complex adaptive system

Copyright © 2005-14 by KESL   18

The profession of project management

- Profession:
  – Body of knowledge
  – Code of ethics
  – Education
  – Certification, etc…
  – "Belonging", -> culture?
- Project Management Institute (PMI)
  – And a handful of others: IPMA, APM (UK)..

Project Management
Body of Knowledge (PMBOK)

- PMBOK: Developed by the Project Management Institute (PMI) (USA)
- Initial publication in 1987; republished in 1996
- Adopted by IEEE as IEEE 1490-1998
  – *Wow, that was a bad move for software (I think)*
- NOT software specific

- A certification: Project Management Professional (PMP)
  – And now Agile project certification
- Coming in 2013: Software extension to PMBOK (SWX)

Source: PMI & IEEE

Project Management
Body of Knowledge

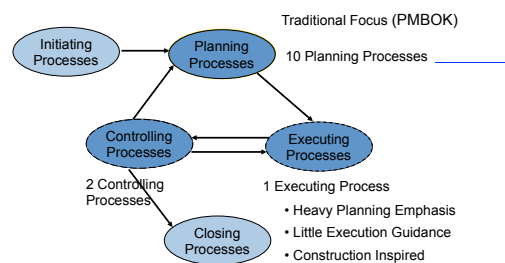| Project Integration Management	Project Scope Management	Project Time Management
Project Plan Development Project Plan Execution Integrated Change Control	Initiation Scope Planning Scope Definition Scope Verification Scope Change Control	Activity Definition Activity Sequencing Activity Duration Estimation Schedule Development Schedule Control
Project Cost Management	**Project Quality Management**	**Project Human Resource Management**
Resource Planning Cost Estimating Cost Budgeting	Quality Planning Quality Assurance Quality Control	Organizational Planning Staff Acquisition Team Development
Project Communications Management	**Project Risk Management**	**Project Procurement Management**
Communications Panning Information Distribution Performance Reporting Administrative Closure	Risk Management Planning Risk Identification Qualitative Risk Analysis Quantitative Risk Analysis Risk Response Planning Risk Monitoring & Control	Procurement Planning Solicitation Planning Solicitation Source Selection Contract Administration Contract Closeout

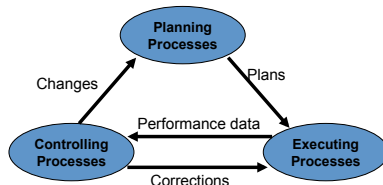Source: E. Lopes Cardozo, Empulsys

Looking into the PMBOK Philosophy

Traditional Focus (PMBOK)

10 Planning Processes _____

2 Controlling Processes

1 Executing Process
- Heavy Planning Emphasis
- Little Execution Guidance
- Construction Inspired

Sources: L. Koskela & Mike Griffiths

Basic Loops in the PMBOK

Sources: L. Koskela & Mike Griffiths

Basic Elements & Theories & Practices

- Planning processes
  – Well-understood and pushed to the extreme in some industries; WBS; Activities/tasks; Risks
  – Estimation (when? how much?)
- Executing processes
  – Task assignment
- Controlling processes
  – Thermostat model, cybernetic model
  – What to measure; assessing progress?
  – Earned value

- This is great stuff, right?

- Let us apply all this to software development, shall we?

Software Project is a Different Beast

- Not a construction or assembly process
- Not an administrative process

- Exploratory: trial and error

- No (or few) pre-defined work breakdown structures (WBS)

- The "nice party" metaphor

Why is Software Different?

1. No fundamental laws of software
   - Cannot validate software on blueprints
   - Need to build and test
2. Extreme modifiability
   - End-user & customer want to exploit this
   - Constant changes
3. No manufacturing costs, no border
   - All costs are in design, no economy of scale
4. Technology churn

Summary

- This course is not on project management, but on software project management
- Software is different than other disciplines:
  - Knowledge intensive
  - Creative
  - Not repetitive
  - But software itself is pretty flexible