The Role of Project Management (PM) in Academic Information Technology (IT)

Christopher Brooks, PMP

Center for Hybrid and Embedded Software Systems (CHESS) Executive Director

Feb. 16, 2012, UC Berkeley College of Engineering IT Staff

Christopher Brooks

- I've been doing IT professionally since 1990, my first machine was a Sun 3/160. @ Berkeley since 1991.
- I'm a release engineer, training electrical engineers in the art of software engineering.
- I've worked with Professor Edward A. Lee since 1992
- I helped develop the Gigascale website in PhP (1999)
- In 2009, I became a Project Management Professional (PMP) via UC Extension
  - 11 classes over 4.5 years, 330 hours of lecture
- These days, I spent ~10% on Sysadmin and Web

Software Projects Success/Challenged/Impaired

- Standish Group publishes reports where projects are surveyed and classified into three categories:

"Resolution Type 1, or **project success**. The project is completed on time and on budget, offering all features and functions as initially specified."

"Resolution Type 2, or **project challenged**. The project is completed and operational but over budget and over the time estimate, and offers fewer features and functions than originally specified."

"Resolution Type 3, or **project impaired**. The project is cancelled at some point during the development cycle."

Standish Group Report, "Chaos," 1995. http://www.projectsmart.co.uk/docs/**chaos-report**.pdf

Why Projects Fail: Standish Group

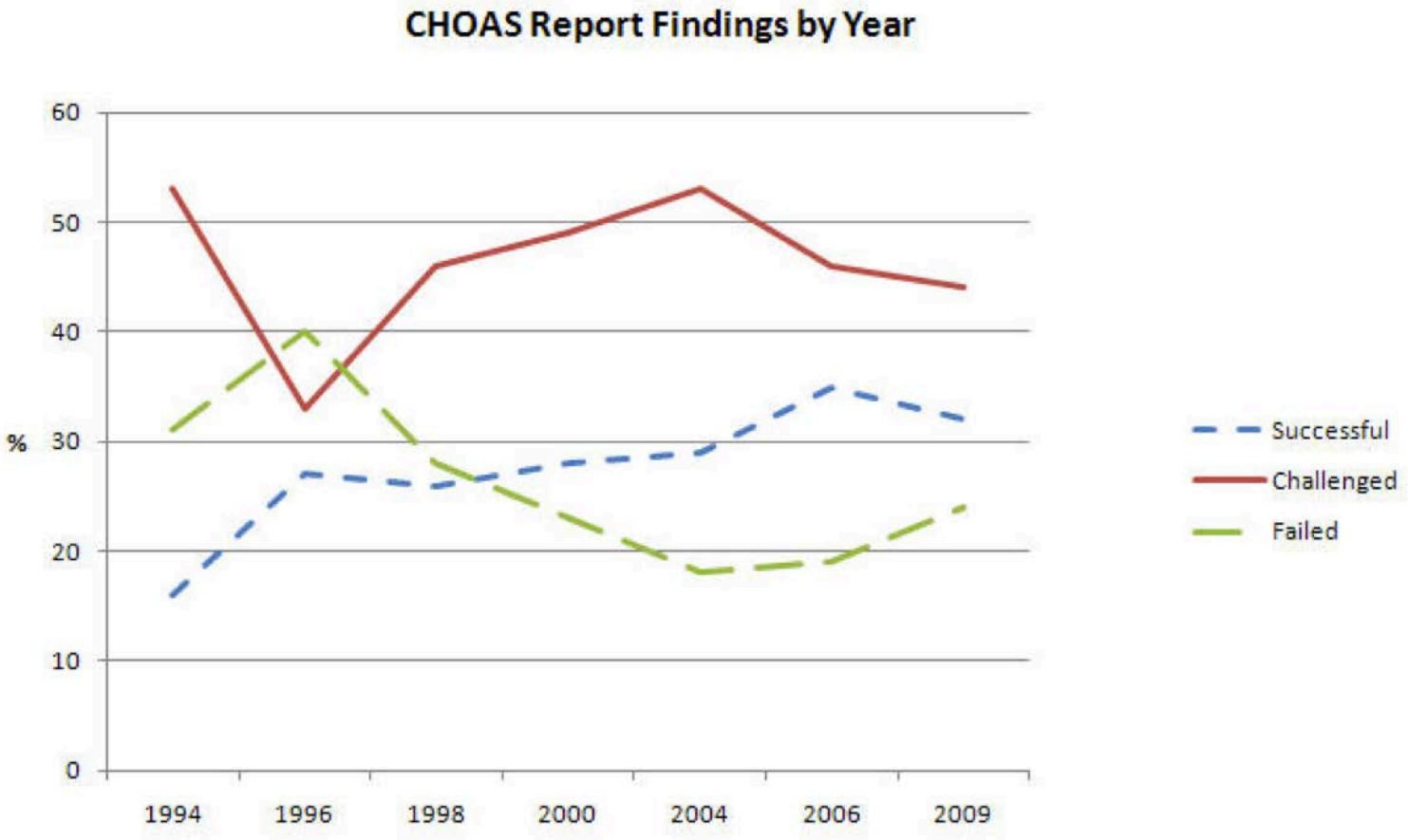


Figure from http://leadinganswers.typepad.com/leading_answers/2011/04/lies-damn-lies-and-statistics.html, based On Eveleens & Verhoef "The Rise and Fall of the Chaos Report Figures," IEEE Software, January/February 2010

Software Project Failures

- Standish Group results are widely reported
  - NITRD: To the US President
- Eveleens & Verhoef refute the Standish Group
  - Definitions: “solely based on an estimation of accuracy for cost, time, and functionality”
  - “one-sided because they neglect underruns for cost and time and overruns for the amount of functionality.”
  - “The organization adopted the Standish definitions to establish when projects were successful. This caused project managers to overstate budget requests to increase the safety margin for success. However, this practice perverted forecast quality.”
  - “Comparing all case studies together, we show that without taking forecasting biases into account, it is almost impossible to make any general statement about estimation accuracy across institutional boundaries.”

http://www.guerrillaprojectmanagement.com/the-chaos-report-myth-busters

Eveleens & Verhoef “The Rise and Fall of the Chaos Report Figures,” IEEE Software, January/February 2010

However,Software Project Failures are Real:

YEAR	COMPANY	OUTCOME (COSTS IN US $)
2005	Hudson Bay Co. [Canada]	Problems with inventory system contribute to $33.3 million* loss.
2004–05	UK Inland Revenue	Software errors contribute to $3.45 billion* tax-credit overpayment.
2004	Avis Europe PLC [UK]	Enterprise resource planning (ERP) system canceled after $54.5 million† is spent.
2004	Ford Motor Co.	Purchasing system abandoned after deployment costing approximately $400 million.
2004	J Sainsbury PLC [UK]	Supply-chain management system abandoned after deployment costing $527 million.†
2004	Hewlett-Packard Co.	Problems with ERP system contribute to $160 million loss.
2003–04	AT&T Wireless	Customer relations management (CRM) upgrade problems lead to revenue loss of $100 million.
2002	McDonald's Corp.	The Innovate information-purchasing system canceled after $170 million is spent.
2002	Sydney Water Corp. [Australia]	Billing system canceled after $33.2 million† is spent.
2002	CIGNA Corp.	Problems with CRM system contribute to $445 million loss.
2001	Nike Inc.	Problems with supply-chain management system contribute to $100 million loss.
2001	Kmart Corp.	Supply-chain management system canceled after $130 million is spent.
2000	Washington, D.C.	City payroll system abandoned after deployment costing $25 million.
1999	United Way	Administrative processing system canceled after $12 million is spent.
1999	State of Mississippi	Tax system canceled after $11.2 million is spent; state receives $185 million damages.

Source: Robert N. Charette, “Why Software Fails”, IEEE Spectrum, September, 2005
http://spectrum.ieee.org/computing/software/why-software-fails

FBI: Virtual Case File (2000-05): $105,000,000.00

9/2000: FBI Announces Trilogy: modern desktops, update LAN, modernize software (Virtual Case File (VCF))

6/01: Cost-plus contract to SAIC

12/01: Scope changed: Replace all apps. Use Oracle. (1st envelope?)

12/02: Get more funding. (2nd envelope?)

2003: Three CIO's before Zal Azmi. (3rd envelope?)

12/03: SAIC Delivers a version of ACF. FBI says its inadequate.

4/04: FBI Director says it will be ready by 6/06.

5/04: SAIC needs $50m, gets $12m. Aerospace Corp gets $2m

1/05: Aerospace's report says to abandon the project.

1/05: VCF scrapped at a cost of $105m

3/05: New project: Sentinel: Lockheed Martin ($424m), still going on.

Sources: Wikipedia. Eggen, Witte, "The FBI's Upgrade That Wasn't", Washington Post, 8/18/06.
Goldstein, "Who Killed the Virtual Case File." IEEE Spectrum, 9/05
http://spectrum.ieee.org/computing/software/who-killed-the-virtual-case-file

FBI: Virtual Case File
Office of the Inspector General (OIG)

Started Contract without:

- "defined requirements,"
- "specific milestones,"
- "critical decision review points, and"
- "penalties for poor contractor performance."

8 causes for problems with the project:

1. "Poorly defined and slowly evolving design requirements"
2. "Contracting weaknesses: Weak statement of work"
3. "IT investment management weaknesses"
4. "Lack of an Enterprise Architecture"
5. "Lack of management continuity and oversight"
6. "Unrealistic scheduling of tasks"
7. "Lack of adequate project integration"
8. "Inadequate resolution of issues raised in reports on Trilogy"

Root causes of failures

Project management	IT	Software Methodology
X		
X		
	X	
?	X	
X		
X		
?	?	X
X		

Source: Congressional Testimony, US DOJ Inspector General Glenn A. Fine, February 2005
http://www.usdoj.gov/oig/testimony/0502/final.pdf

Software Project Failure Factors

- Unrealistic or unarticulated project goals
- Inaccurate estimates of needed resources
- Badly defined system requirements
- Poor reporting of the project's status
- Unmanaged risks
- Poor communication among customers, developers, and users
- Use of immature technology
- Inability to handle the project's complexity
- Sloppy development practices
- Poor project management
- Stakeholder politics
- Commercial pressures

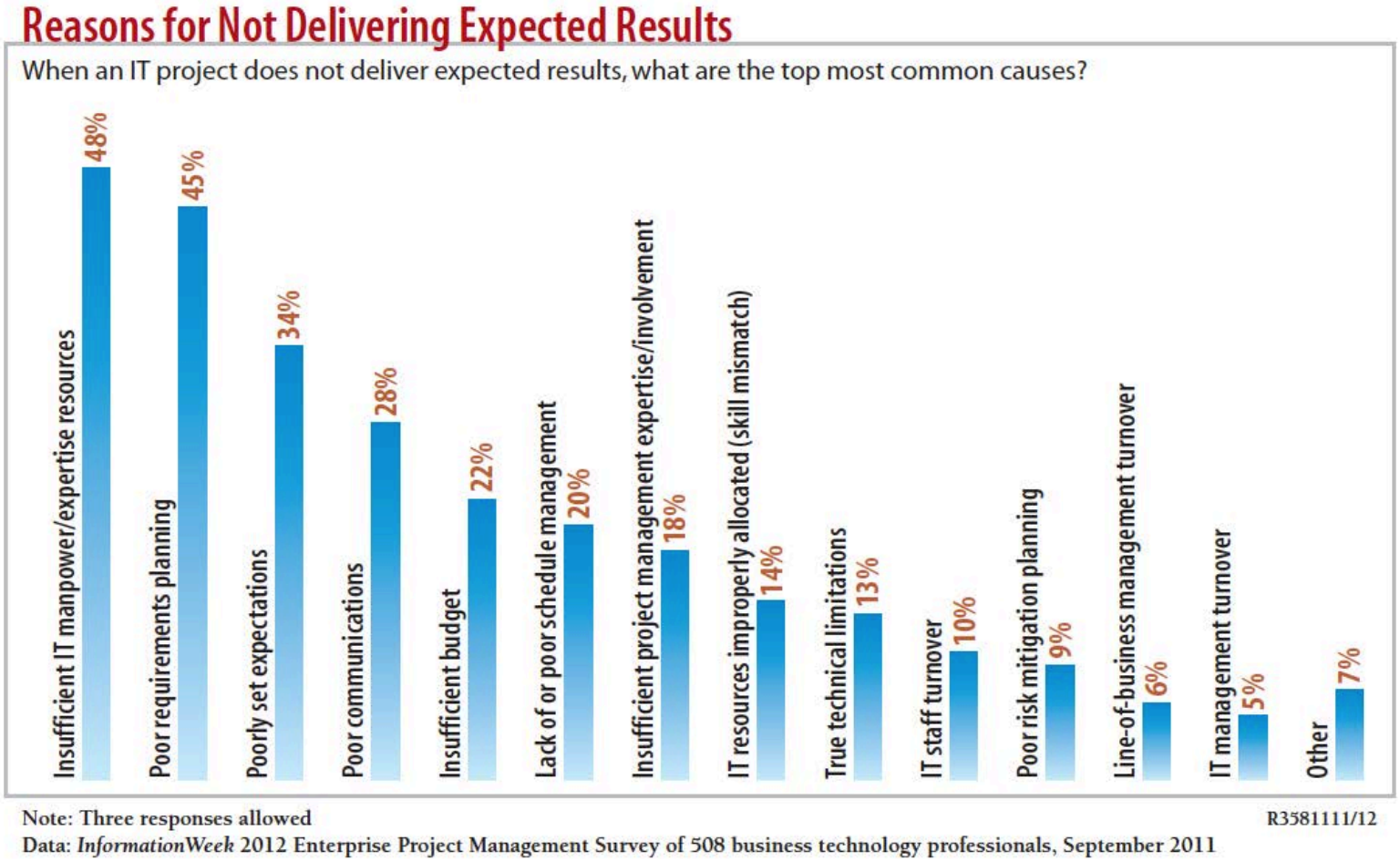
Source: Robert N. Charette, “Why Software Fails”, IEEE Spectrum, September, 2005
http://spectrum.ieee.org/computing/software/why-software-fails

2011 Enterprise IT Project Management Data

- 30% of “respondents say IT projects almost always deliver value to the business” (Survey of 508 techs in 9/11)
- The lean model (fail fast) is seen as a solution. (more later)
- Project Management Offices are seen as a problem.
- 58% use formal project management methodologies, down from 70% in 2010. (Survey of 508 techs in 9/11, 684 in 6/10)

Source: Jonathan Feldman, “Project Management Gets Lean,” Information Week, 2/13/2012
http://informationweek.com/1324/projectmanagement

The top most common cause for not delivering expected results in an IT project



Source: Jonathan Feldman, “Project Management Gets Lean,” Information Week, 2/13/2012
http://informationweek.com/1324/projectmanagement

Why do IT projects fail?

"Managing an IT project is like juggling chunks of Jell-O: It's neither easy nor pretty. Information technology is especially slippery because it's always moving, changing, adapting and challenging business as we know it."

IT Projects fail because:

- They are harder: technology changes, hardware/software
- They fail at the beginning: lack of planning & resources
- They fail because they are rushed: Must. Remain. Competitive.
- They fail because of unwieldy scope: too large

Source: (Marti Hearst used the quote in her 2/8/2012 Citris Talk)
Joseph Phillips, "**Project Management Definition and Solutions," CIO Magazine**
http://www.cio.com/article/40342/Project_Management_Definition_and_Solutions

Software and IT Projects fail…

because of Project Management failures

So, let's look at traditional Project Management and at newer Project Management techniques.

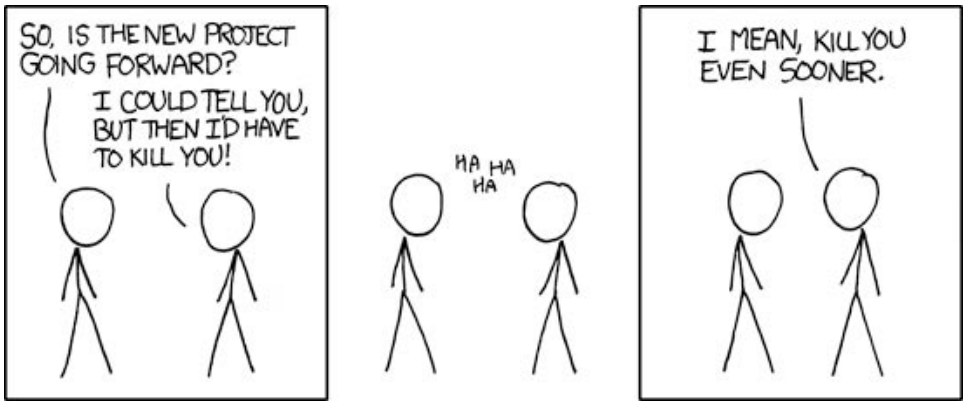
PMI: Project Management Institute

- "*A Guide to the Project Management Body of Knowledge* " (aka PMBOK)

- PMI Certification
  - *Certified Associate in Project Management (CAPM)*
  - *Project Management Professional (PMP)*
  - *Program Management Professional (PgMP)*

Why does Project Management Matter?

- Many efforts are project based:
  - Movies, electronic hardware, construction, weddings
- You will meet project managers and it is good to know when you being project managed.
- There are always new methodologies and technologies, one can spend all of one's time learning and never actually doing.
- So you know when Bad Things™ are about to happen:



Credit: xkcd.com

What is a project?

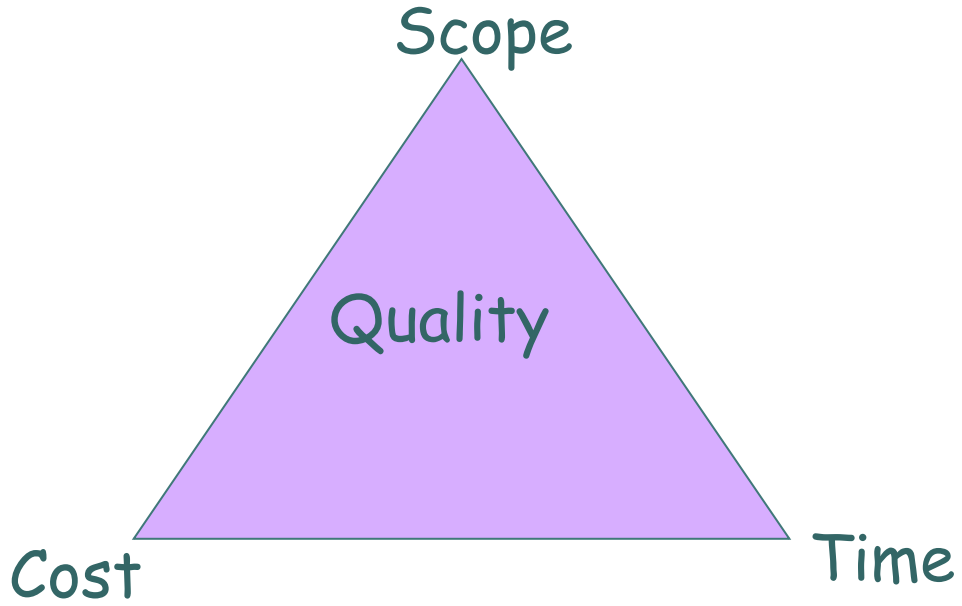
Q: What is a project?

A: "A temporary endeavor undertaken to create a unique *product*, *service* or *result*" -source PMBoK, p368

Note that a project is a **temporary** endeavor with a start and a finish. Avoid confusing continuous processes with projects.

Project Management

- Project management is?
- The art of managing projects to a successful completion.
- The Project Management Triple Constraint:



Time, Cost, Scope and Quality in the definition of a Project

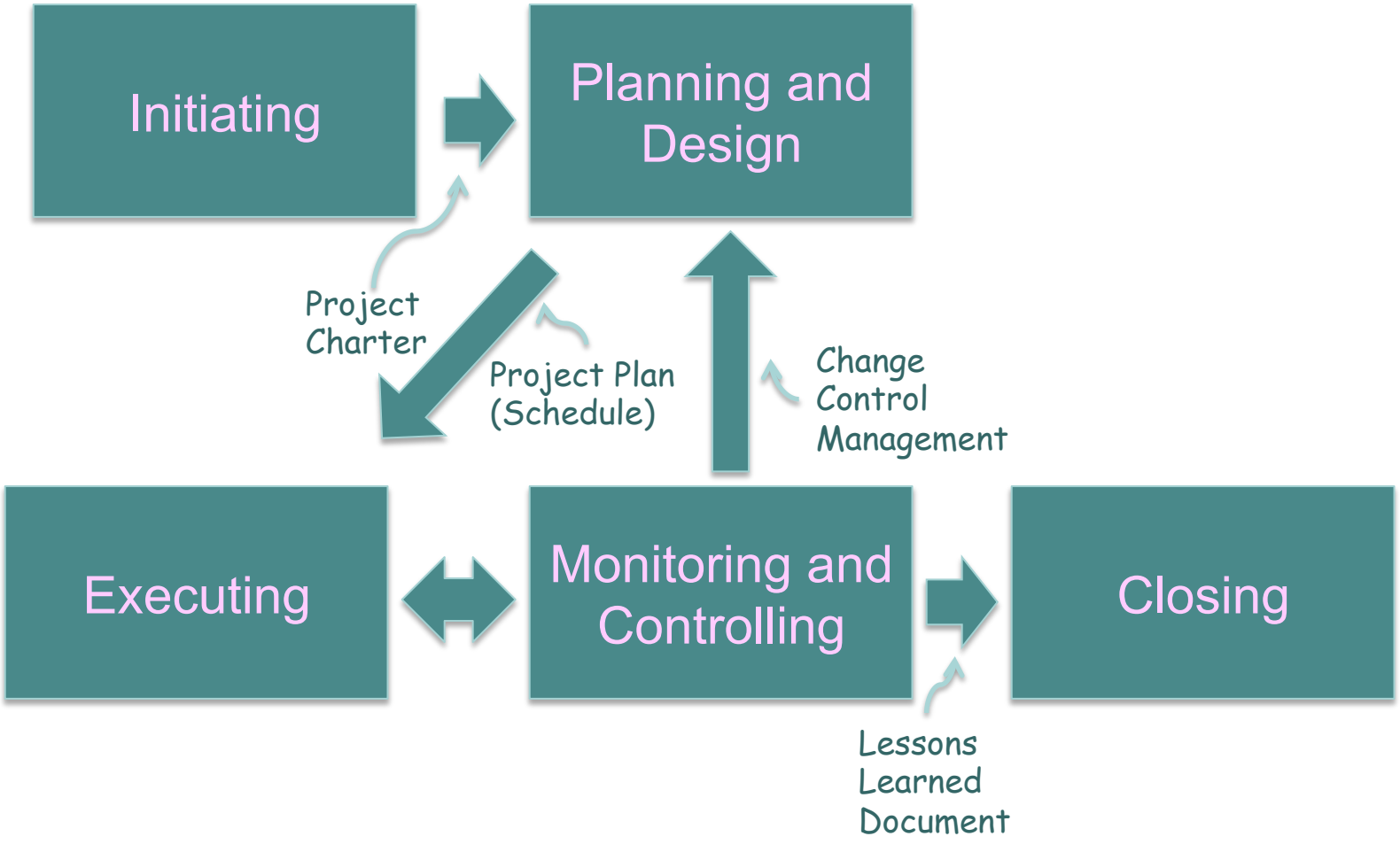
Q: What is a project?

A: "A temporary endeavor undertaken to create a unique *product*, *service* or *result*"

Where are time, cost, scope and quality?

Quality is missing from the definition!

Project Management Phases



How to successfully manage a project

- Develop a **one** page project charter
  - Get your sponsor to sign off
- Develop a time line with milestones
  - Work backwards
  - Describe deliverables
  - Break up deliverables into smaller deliverables
  - Assign deliverables to specific people
  - Associate Milestones with Deliverables
- Monitor progress
  - If you don't look at the plan, then why bother?
  - Status reports can be email messages or meetings

Project Charter

- Formalize the project with the sponsor
- Sections:
  - Project Overview
  - Project Approach
  - Project Objectives
  - Major Deliverables
  - Constraints
  - Risks and Feasibility
- In one page
  - Most people, especially busy people, will not read more. In fact, it has been shown that most people don't even read (hi mom!) slides.

Project Charter Example: Web Site

One page!

www-cad Project Overview

This project is to create a new web site for the CAD group faculty. The current website at www-cad.eecs has a very old look, it needs an update so that we can attract new students.

Project Approach

The project is a fairly small website based partly on a preexisting site, so we will use a classic waterfall approach with milestones. The project team will consist of the following people. I've estimated the maximum amount of time we can get from each person over the life of the project.

Kurt Keutzer (2 hrs week for 6 weeks) Ken Lutz (2 hrs/week for 6 weeks)
Brad Krebs (10 hrs/week for 6 weeks) Christopher Brooks (10 hrs/week for 6 weeks)
Allen Hopkins (5 hrs/week for 6 weeks) Carol Sitea (1 hr/week for 6 weeks)

The project sponsor is Professor Keutzer. Professor Keutzer is on sabbatical this semester, but we hope to get feedback from him on a continuing basis.

Project Objectives

- Update the look and feel of the website to a modern standard
- Provide access to student and faculty pages
- Provide access to active projects
- Provide access to summaries, downloads and key papers of inactive projects. The old pages of inactive projects should be archived.
- Provide a simple static listing of seminars. A more complex calendar and a search engine are deferred due to schedule constraints.

Major Deliverables

- A schedule along with time estimates.
- A prioritized list of features.
- An example of the main page so we can review look and feel.
- An archive of the old website
- The final website.

Constraints

Professor Keutzer would like to see the web site completed by mid-March: that is when students start looking at graduate schools. Developers might not have much time to work on this project. The project requires timely feedback from the faculty.

Risk and Feasibility

The primary risk is that the project takes too long to complete and we miss the mid-March opportunity. Another risk is that we complete the project too quickly and quality suffers. A third risk is that there are only so many resources available. By fast tracking, we can handle some of the tasks in parallel and avoid these risks. The project is definitely feasible if we roll out the website in stages.

Project Charter: Overview

- In a sentence, describe the project:

This project is to create a new web site for the CAD group faculty.

- Possibly include the business reason:

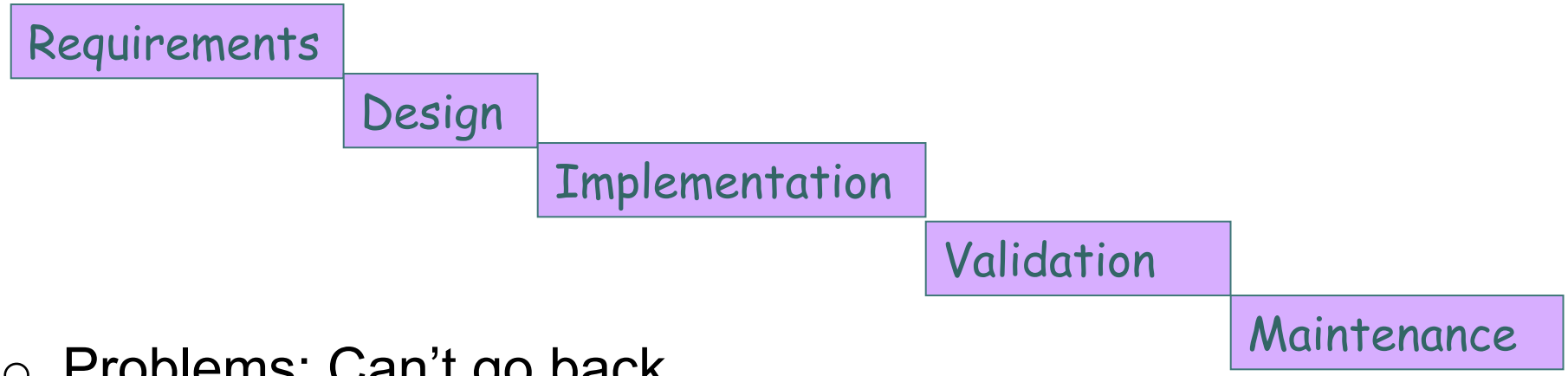
The current website at www-cad.eecs has a very old look, it needs an update so that we can attract new students.

Project Charter: Approach

- Describe your software development life cycle (SDLC):
- What's that?
  - Methodologies: Waterfall, V-Model, Iterative, Spiral, **Agile**, Lean, XP, Cleanroom, Rapid Application Development (RAD), Rational Unified Process (RUP), Scrum, Test Driven Development (TDD)

Waterfall Method

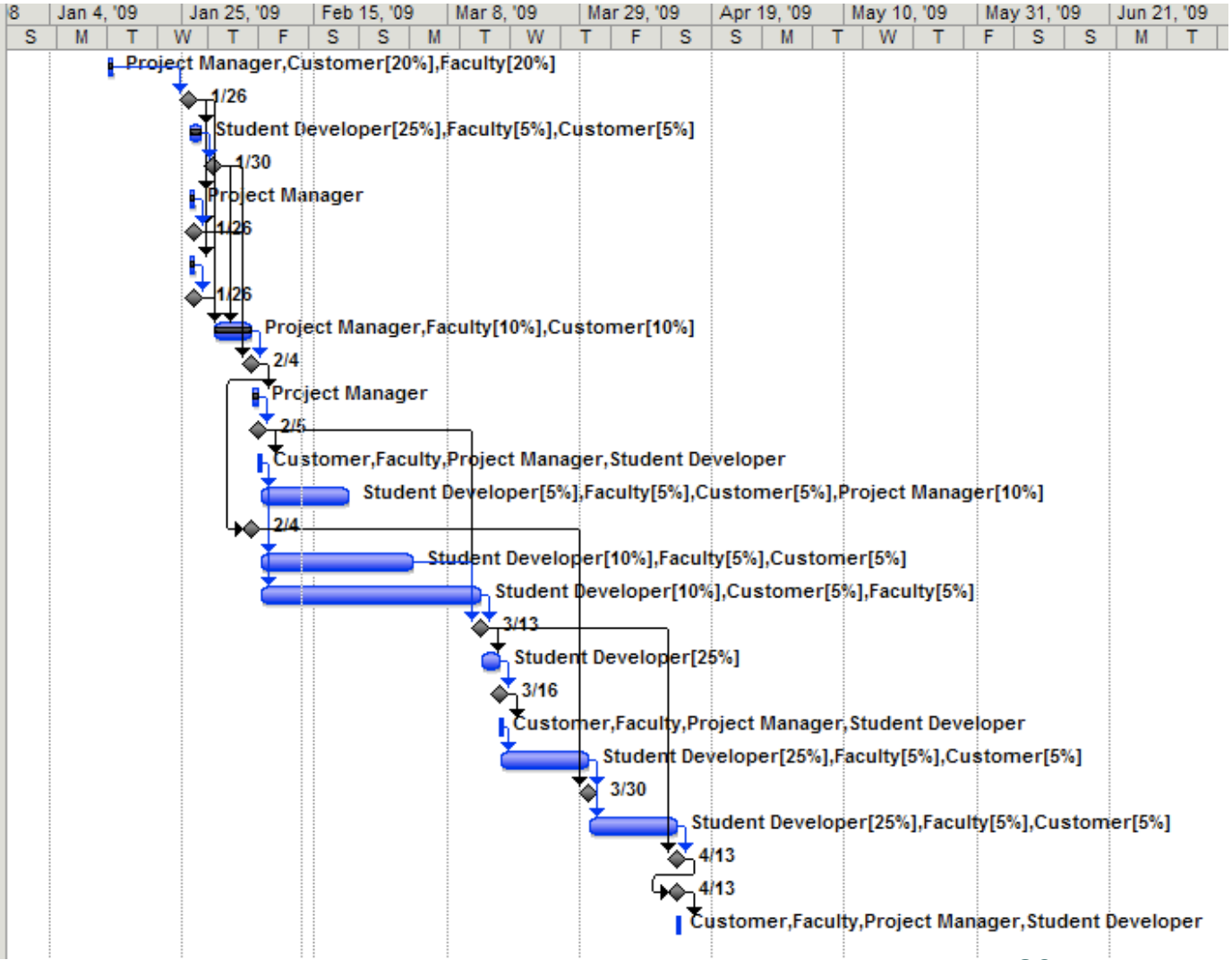
- Complete one phase before going on to the next [Royce, 1970]



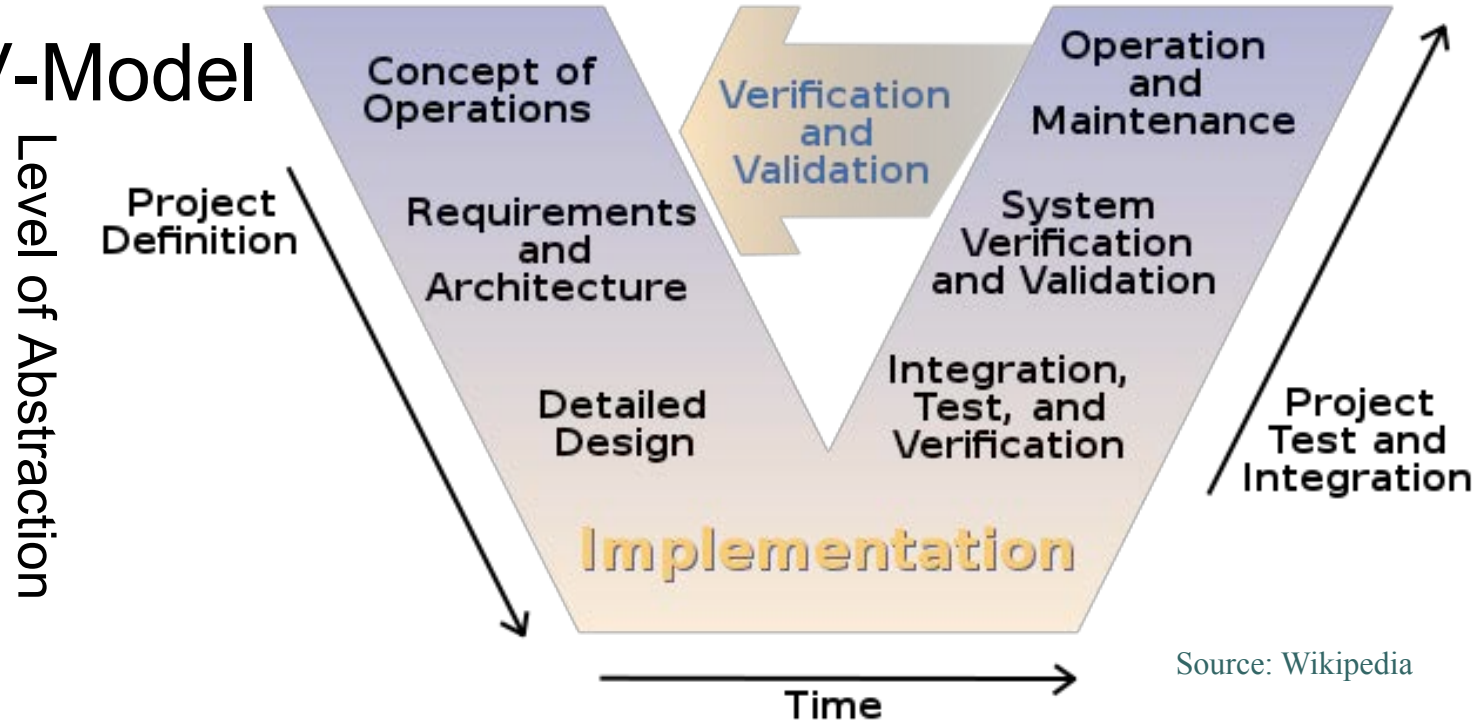
- Problems: Can't go back
- Good for extending known solutions

Waterfall Example: Microsoft Project Gantt Chart

	Task Name	Duration
1	Develop Project Charter	1 day
2	Project Charter	0 days
3	Develop User Interface Design	2 days
4	User Interface Design Doc.	0 days
5	Develop Risk Mgmt Plan	1 day
6	Risk Management Plan	0 days
7	Develop Change Mgmt Plan	1 day
8	Change Management Plan	0 days
9	Develop Project Management Plan	4 days
10	Project Management Plan	0 days
11	Develop Project Schedule	1 day
12	Project Schedule	0 days
13	Kickoff Meeting	1 hr
14	Develop Marketing Plan	10 days
15	Marketing Plan	0 days
16	Develop new features	16.67 days
17	Bug fixes	25 days
18	Beta Software Release	0 days
19	Develop Status Report	1 day
20	Status Report	0 days
21	Status Meeting	1 hr
22	Develop website & poster	10 days
23	Marketing Website & Poster	0 days
24	Bug Fixes	10 days
25	Final Software & Doc Release	0 days
26	Lessons Learned Doc.	0 days
27	Closing Meeting	1 hr



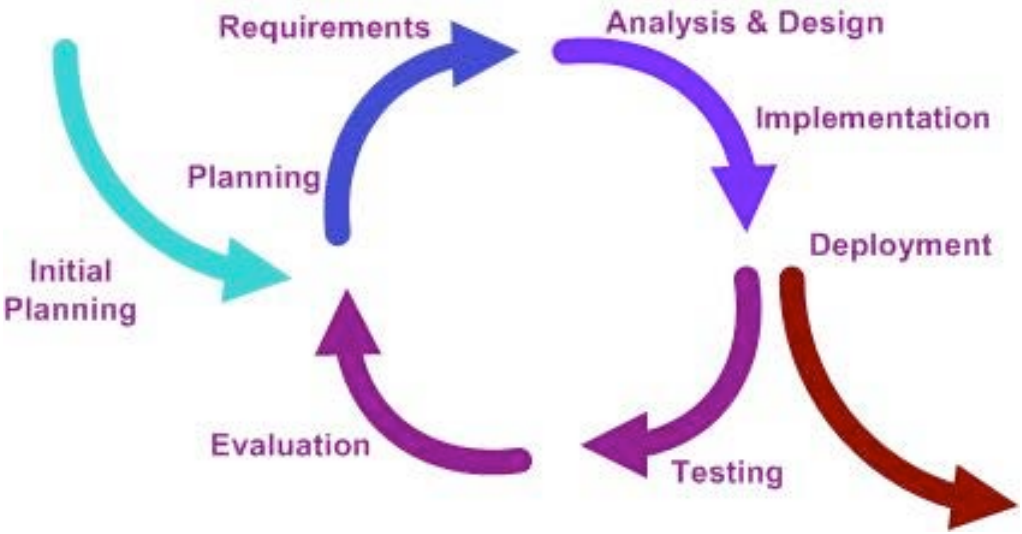
V-Model



Source: Wikipedia

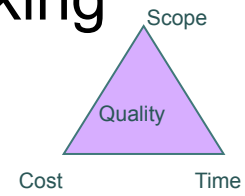
- Based on Waterfall
- Validation: “Are you building the right thing?” (User)
- Verification: “Are you building it right?”
- Shows relationship between each phase of development and its testing phase

Iterative and Incremental Development



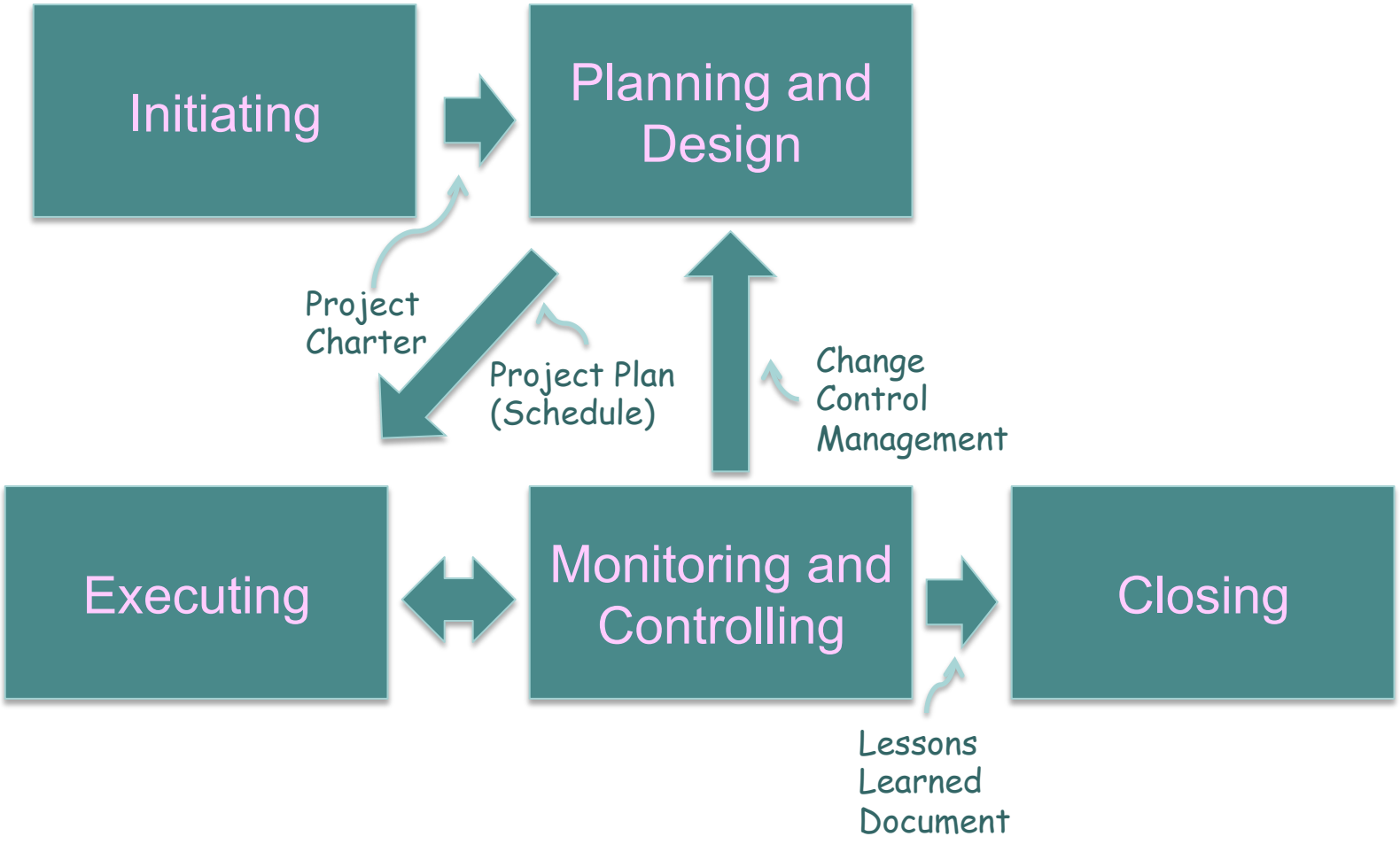
- Project Mercury (1960)
- Three steps
  - Initialization
  - Iteration
  - Project Control List: tasks and new features
- Advantage over Waterfall: Iterative allows backtracking
- Iterative is time boxed, not feature boxed
- Spiral, Agile and RUP are based on Iterative

Source: Wikipedia



Project Management Phases

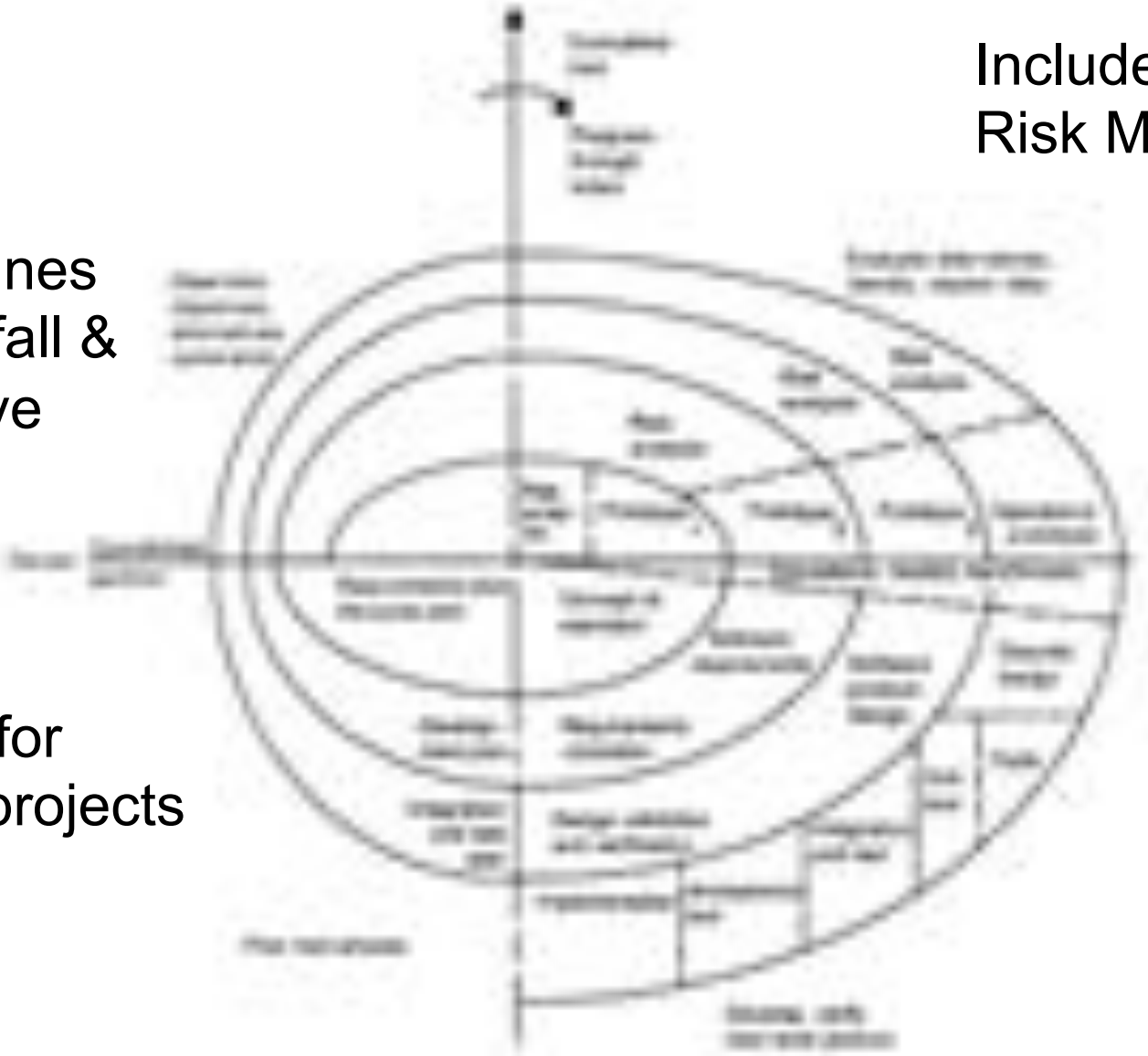
Recall the PMBoK phases and the similarity to Iteration



Spiral Model [Boehm, ’88]

Includes
Risk Management

Combines
Waterfall &
Iterative



Good for
large projects

- Source: B.W Boehm, “A spiral model of software development and enhancement,”

Agile Development

- A group of methodologies
- Based on Iterative/Incremental
- Break up tasks
  small timeboxes (1-4 weeks)
- Each iteration: Full devel cycle:
  Planning, Requirements, Design, Coding, Testing
- Customer Representative:
  Appointed by stakeholders. Answers Questions
- Adaptive, not Plan-driven

Source: Dbenson/VersionOne CC SA3.0

Manifesto for Agile Software Development

"We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:"

"Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan"

"That is, while there is value in the items on the right, we value the items on the left more."

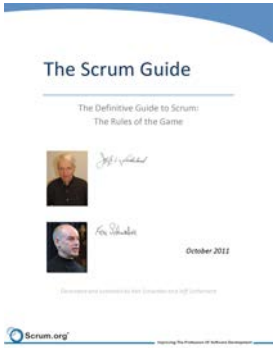
Source: http://agilemanifesto.org/ (Beck et. al, 2001)

Lean Software Development

- Based on a the Toyota Production System (1948-1975), originally known as Just In Time (JIT) production
- The Toyota Way: Continuous Improvement and **Respect** for People
- Lean has 7 Principles:
  - Eliminate Waste: Remove unnecessary code, requirements
  - Amplify Learning: Short cycles. Feedback from customer
  - Decide as late as possible: Max. Flexibility. Wait for customer changes
  - Deliver as fast as possible: Frequent Releases. Parallel Teams
  - Empower the team: Developer access to customer…
  - Build integrity in: Unit/Integration/System/Acceptance Tests. Refactoring.
  - See the whole: "Think big, act small, **fail fast**; learn rapidly"

Scrum

Wikipeida:
cc sa3.0



Source: The Scrum Guide,
Schwaber & Sutherland (2011)

- Scrum: Rugby: Restarting the game
- "Scrum: A framework within which people can address complex adaptive problems, while productively and creatively delivering products of the highest possible value. Scrum is:
  - Lightweight
  - Simple to understand
  - Extremely difficult to master"
  - Scrum has different components:
    - Roles, Artifacts, Events

Scrum Roles

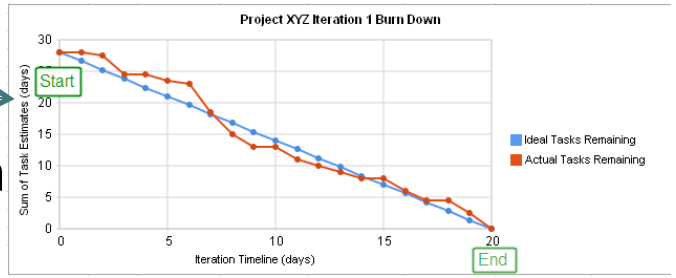
- Scrum Master: ensures process is followed, removes impediments
- Product Owner: Represents stakeholders
  - Not the Project Manager
- Development Team: The people (7 +/-2) doing the work
  - No titles other than “Developer”
  - The development team decides how to do the work

Scrum Artifacts: Product Backlog

- Product Backlog
  - Everything needed in the product
  - Product Owner is responsible for content
  - An ordered list of items.
    - Each item description, order, and estimate
    - Ordered by value, risk, priority, and necessity
  - Product Backlog grooming: add detail, update estimates, change order. Usually no more than 10% of development team's capacity
  - Monitoring: Total work remaining is visible.

Scrum Artifacts: Sprint Backlog & Increment &"Done"

- Sprint Backlog: the set of Product Backlog Items (PBI)s for the current sprint and a plan to complete the sprint goal.
  - Level of detail: Enough to show changes in progress during the Daily Scrum (2 hour <= t <= 2 days)
  - Monitoring: Total work remaining is visible.
- Interference list
- Task Board: Ideally physical board: Started/In Progress Completed.
- Burn Down Chart: Hours remaining, update when PBIs are done.
- Increment: Sum of all PBIscompleted during th current sprint an previous sprints.
- Definition of "Done": A shared understanding of what has been completed



Burn Down Chart Image: i8Bug, Wikipedia, CC BY-SA 3.0

Scrum Events

- Sprint: 1-5 weeks
- Daily Scrum: Fast stand up meeting: What was done yesterday? What will be done today? Am I blocked or will I be blocked? (15min)
- Backlog grooming: estimate existing backlog (5% of sprint)
- Scrum of Scrums: Each group sends a representative
- Sprint Planning: estimate product backlog, prepare the sprint backlog (Planning Poker™) (4h)
- Sprint Review: (end of sprint). Present to stakeholders (4h)
- Sprint Retrospective: (end of sprint). What worked? (3h)

Scrum: Pros and Cons

PROS

- Daily Scrum is efficient: hard to avoid work
- Scrum works best if everyone is 100% on one project
- Scrum works well if everyone is physically co-located
- Development teams are 5-9 people + scrum master and product master

Cons

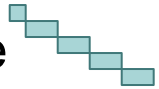
- People have multiple projects
- Harder to use with virtual teams (though not impossible)
- Fewer than 5-9 people do not get much done

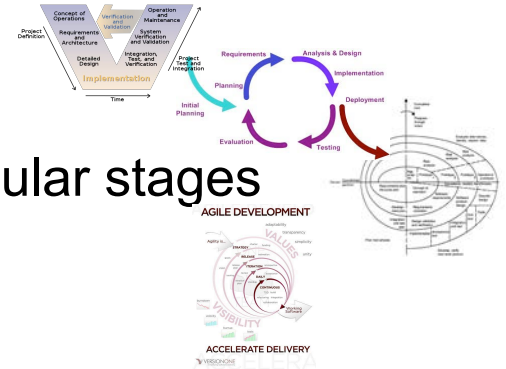
Extreme Programming (XP)

- Planning
  - Release planning
  - Frequent Releases
- Coding
  - Standards
  - Unit tests first
  - Pair Programming
  - Late Optimization
- Designing
  - Simplicity
  - System Metaphor
  - Refactoring
- Testing
  - Unit Tests
  - Bugs need Tests
  - Acceptance Tests

Based on http://www.extremeprogramming.org/rules.html

Software Development Life Cycles (SDLCs)





- Waterfall - Complete a phase before the next phase
- V-Model – Development vs. testing
- Iterative – Cyclic phases – Basis for many SDLCs
- Spiral – Waterfall & Iterative, Risk Management at regular stages
- Agile
  - Lean: Based on Just In Time. 7 principles
  - Scrum: Fast standup meeting: What was done yesterday? What will be done today? Am I blocked or will I be blocked?
  - Test-Driven Development (TDD): Write a Unit Test before coding
  - Extreme Programming (XP): **Fine scale feedback** (pair programming, TDD), **Continuous Process** (small releases), **Shared Understanding** (Coding standards), **Programmer welfare** (sustainable pace)
- Cleanroom: Formal Methods, Statistical Quality Control, Statistically Sound Testing
- Rapid Application Development (RAD): Construct Prototypes
- Rational Unified Process (RUP): Adaptable Process Framework

Suitability of different SDLC's

Agile Home Ground ***Adaptive methods*** **(Lean, Scrum, XP …)**	**Plan-driven Home Ground** ***Predictive Methods*** **(Waterfall, Iterative, Spiral …)**	**Formal Methods (Cleanroom …)**
Low Criticality	High Criticality	Extreme Criticality
Senior Developers	Junior Developers	Senior Developers
Requirements change often	Requirements do not change often	Limited requirements, Limited Features (Wirth's Law[1])
Small Number of Developers	Large Number of Developers	Requirements that can be modeled
Culture that thrives on chaos	Culture that demands order	Extreme quality

Based on: Wikipedia Agile Article, From Boehm and Turner (2004)

[1]Wirth's Law: "Software is getting slower more rapidly than hardware becomes faster" (1995)

Project Charter: Approach (again)

- Describe your software development life cycle (SDLC):
- What's that?
  - Methodologies: Waterfall, V-Model, Iterative, Spiral, **Agile**, Lean, XP, Cleanroom, Rapid Application Development (RAD), Rational Unified Process (RUP)
  - Techniques: Scrum, Test Driven Development (TDD)
- List the team
- List the time commitments

Project Charter: Approach Example

The project is a fairly small website based partly on a preexisting site, so we will use a classic waterfall approach with milestones. The project team will consist of the following people. I've estimated the maximum amount of time we can get from each person over the life of the project.

Kurt Keutzer (2 hrs week for 6 weeks)

Ken Lutz (2 hrs/week for 6 weeks)

Brad Krebs (10 hrs/week for 6 weeks)

Christopher Brooks (10 hrs/week for 6 weeks)

Allen Hopkins (5 hrs/week for 6 weeks)

Carol Sitea (1 hr/week for 6 weeks)

The project sponsor is Professor Keutzer. Professor Keutzer is on sabbatical this semester, but we hope to get feedback from him on a continuing basis.

Project Charter: Objectives

- List your objectives
- Objectives are one way to measure success
- Don't get too specific, there is time for that later
- Check your objectives with customer provided documentation
  - Are you solving the problem presented by the customer?

Project Charter: Objectives Example

- Update the look and feel of the website to a modern standard
- Provide access to student and faculty pages
- Provide access to active projects
- Provide access to summaries, downloads and key papers of inactive projects. The old pages of inactive projects should be archived.
- Provide a simple static listing of seminars. A more complex calendar and a search engine are deferred due to schedule constraints.

Project Charter: Deliverables

- What's a deliverable?
- Physical artifacts which describe your progress and includes your product
- Deliverables should include:
  - A description (Scope)
  - A date (Time)
  - A person or persons who are responsible (Cost)

Project Charter: Deliverables Example

- A schedule along with time estimates.
- A prioritized list of features.
- An example of the main page so we can review look and feel.
- An archive of the old website
- The final website.

Project Charter: Constraints

- Schedule, Budget and Resource problems
- Example:

Professor Keutzer would like to see the web site completed by mid-March: that is when students start looking at graduate schools. Developers might not have much time to work on this project.
The project requires timely feedback from the faculty.

Project Charter: Risks

- List things that could go wrong and how you will avoid them
- Don't skip the risks.
- Example:

> The primary risk is that the project takes too long to complete and we miss the mid-March opportunity. Another risk is that we complete the project too quickly and quality suffers. A third risk is that there are only so many resources available. By fast tracking, we can handle some of the tasks in parallel and avoid these risks. The project is definitely feasible if we roll out the website in stages.

Why Milestones

- How will you measure success?
- How will you divide up the work?
- How will you make sure everyone participates?

- Milestones

What is a milestone?

- A milestone is a checkpoint in the project
  - A milestone has
    - A description of the point
    - A date
    - A person or persons who are responsible
- A milestone may or may not have a deliverable associated with it.
  - A deliverable has
    - A description of a deliverable (scope)
    - A date (time)
    - A person or persons who are responsible (cost)

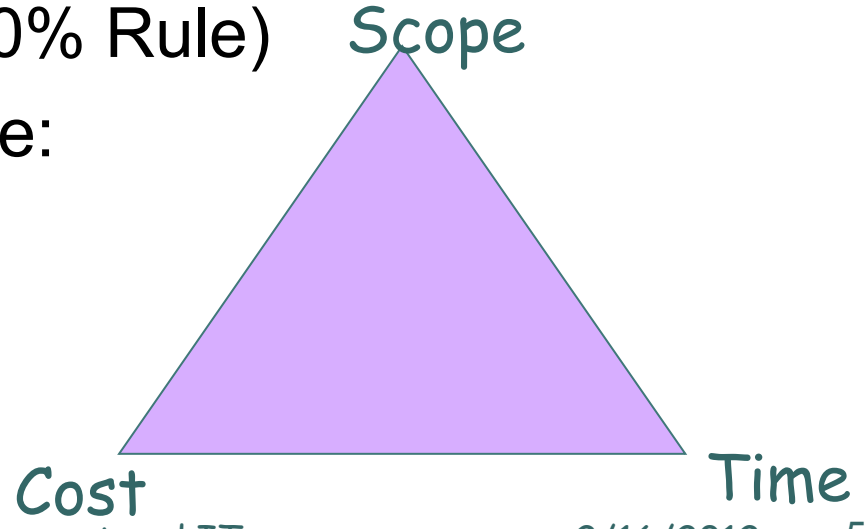
Work Breakdown Structure (WBS)

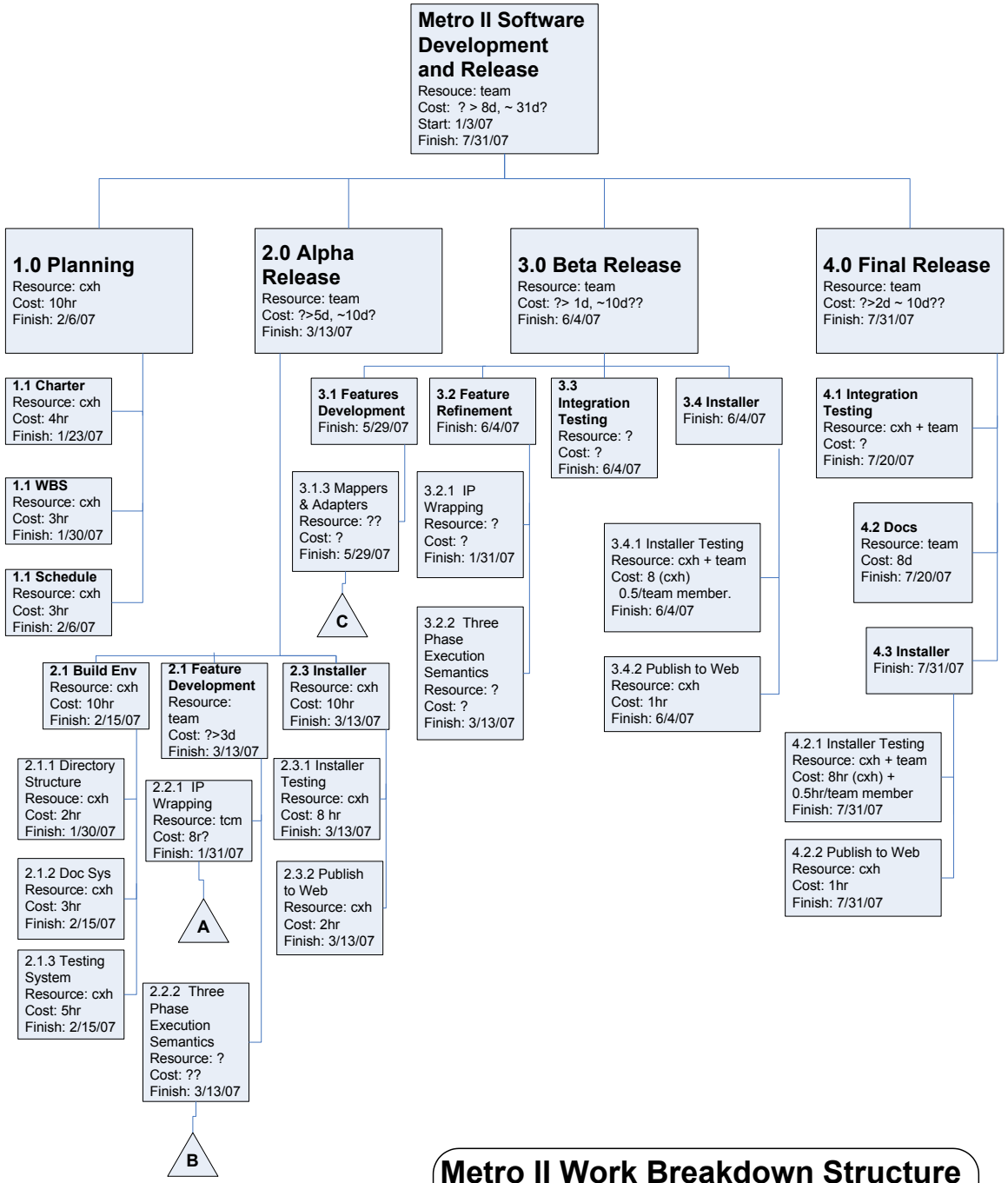
- How to do a schedule: Use a Work Breakdown Structure
- A work breakdown structure is an outline that describes the deliverables.
- Each level of the outline describes 100% of the work below it (the 100% Rule)
- A WBS should include:

  What (Scope)

  When (Time)

  Who (Cost)





Metro II Work Breakdown Structure

Page 1 of 2	3/6/2008	Christopher Brooks

Textual WBS for a small project

WBS	Name	Duration	Effort	Resources	Cost	Finish
spccdt1	Develop Project Charter	1 day	11.2 hrs	PM,Cust,Fac	$1,168	1/13/09
spccdt2	Project Charter	0 days	0 hrs		$0	1/26/09
spccdt3	Develop User Interface Design	2 days	5.6 hrs	Dev,Fac,Cust	$304	1/27/09
spccdt4	User Interface Design Doc.	0 days	0 hrs		$0	1/30/09
spccdt5	Develop Risk Mgmt Plan	1 day	8 hrs	PM	$800	1/26/09
spccdt6	Risk Management Plan	0 days	0 hrs		$0	1/26/09
spccdt7	Develop Change Mgmt Plan	1 day	0 hrs		$0	1/26/09
spccdt8	Change Management Plan	0 days	0 hrs		$0	1/26/09
spccdt9	Develop Project Mgmt Plan	4 days	38.4 hrs	PM,Fac,Cust	$3,936	2/4/09
spccdt10	Project Management Plan	0 days	0 hrs		$0	2/4/09
spccdt11	Develop Project Schedule	1 day	8 hrs	PM	$800	2/5/09
spccdt12	Project Schedule	0 days	0 hrs		$0	2/5/09
spccdt13	Kickoff Meeting	1 hr	4 hrs	Cust,Fac,PM,Dev	$360	2/6/09
spccdt14	Develop Marketing Plan	10 days	10.22 hrs	Dev,Fac,Cust,PM	$789	2/20/09
spccdt15	Marketing Plan	0 days	0 hrs		$0	2/4/09
spccdt16	Develop new features	16.67 days	21.33 hrs	Dev,Fac,Cust	$1,320	3/2/09
spccdt17	Bug fixes	25 days	28 hrs	Dev,Cust,Fac	$1,520	3/13/09
spccdt18	Beta Software Release	0 days	0 hrs		$0	3/13/09
spccdt19	Develop Status Report	1 day	2 hrs	Dev	$60	3/16/09
spccdt20	Status Report	0 days	0 hrs		$0	3/16/09
spccdt21	Status Meeting	1 hr	4 hrs	Cust,Fac,PM,Dev	$360	3/16/09
spccdt22	Develop website & poster	10 days	28 hrs	Dev,Fac,Cust	$1,520	3/30/09
spccdt23	Marketing Website & Poster	0 days	0 hrs		$0	3/30/09
spccdt24	Bug Fixes	10 days	28 hrs	Dev,Fac,Cust	$1,520	4/13/09
spccdt25	Final Software & Doc Release	0 days	0 hrs		$0	4/13/09
spccdt26	Lessons Learned Doc.	0 days	0 hrs		$0	4/13/09
spccdt27	Closing Meeting	1 hr	4 hrs	Cust,Fac,PM,Dev	$360	4/13/09
Totals					$14,817	

Note: Milestones have a duration of 0

WBS Columns Explained

WBS	Name	Duration	Effort	Resources	Cost	Finish
spccdt1	Develop Project Charter	1 day	11.2 hrs	PM,Cust,Fac	$1,168	1/13/09
spccdt2	Project Charter	0 days	0 hrs		$0	1/26/09

- WBS – A unique identifier for the package
- Name
- Duration
  - "The total number of work periods . . . required to complete a schedule activity or work breakdown component" (PMBoK)
- Effort

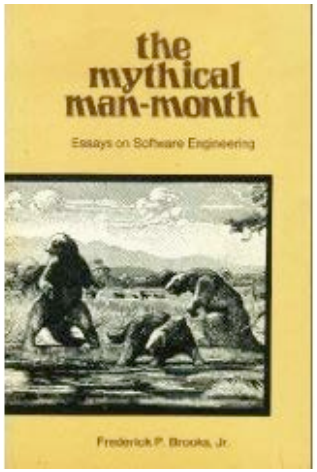
  Q: How many hours per week of effort per person?

  A: Not more than ~32.5 hrs/week
  - "The number of labor units required to complete a schedule activity or work breakdown structure component" (PMBoK)
- Resources
- Cost
- Finish

The Critical Path

- What's the Critical Path?
- The longest path through the project, which determines the shortest time to completion.
- Solutions during Planning:
  - Do things in parallel
  - Add more resources
  - Prune deliverables
- Solutions during Operation:
  - Replan
  - Panic

Brooks's Law



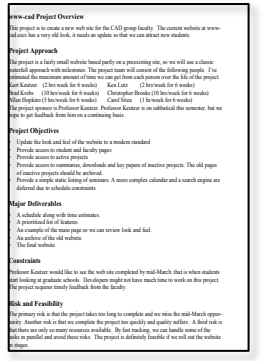
- "adding manpower to a late software project makes it later" Fred Brooks, 1975
- Why?
  - Ramp Up
  - Communication Overhead
- But what about Open Source? (Cathedral and the Bazaar)
  - Cheap Communication
  - Many Programmers

Schedule Problems to Avoid

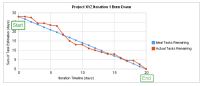
- Milestones are too coarse grained
  - Have at least one milestone per week
  - Why? How will you know if you are falling behind?
- No parallelism in the schedule
  - Why? Get more work done in a shorter term, though integration/synchronization has costs
- Schedule is not updated
  - Why? How will you know if you are falling behind?
- Poor Estimates
  - Why? Humans are optimistic, when was the last time you finished early? One rule of thumb is to multiply an estimate by pi.
    Also, consider that each teammate has other projects.

Project Management Conclusions

- Standish Group: “Lies, damn lies and statistics” (Mark Twain/Disraeli)
- Software Project do fail: FBI ACF “Prepare Three Envelopes”
- Traditional Project Management:
  - Project Management has a triple constraint
  - Preparing a one page charter can help you.
    - One page, defines scope, time, cost, risks
    - Good for working with busy stakeholders and narrowing scope
  - Creating a schedule: Defining milestones and working backwards can help
    - Work Breakdown Structure (WBS) can help



- Software Development Methodologies: Waterfall -> Iterative -> Agile
- Agile – fully buzzword compliant: Time Boxed, Adaptive
- Scrum – Daily Scrum (Done, to be done, in the way)
  - Roles: Product Owner, Scrum Master, Development Team
  - Events: Sprint, Daily Scrum, Sprint Planning, Sprint Review, Sprint Retrospective
  - Artifacts: Product backlog, Sprint backlog, Burn Down Chart.



Questions?

Back up slides

Principles behind the Agile Manifesto

We follow these principles

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

Source: http://www.agilemanifesto.org/principles.html (Beck et. al, 2001)