

Comp8081

Management Issues in Software Engineering

Donna Turner



Agenda

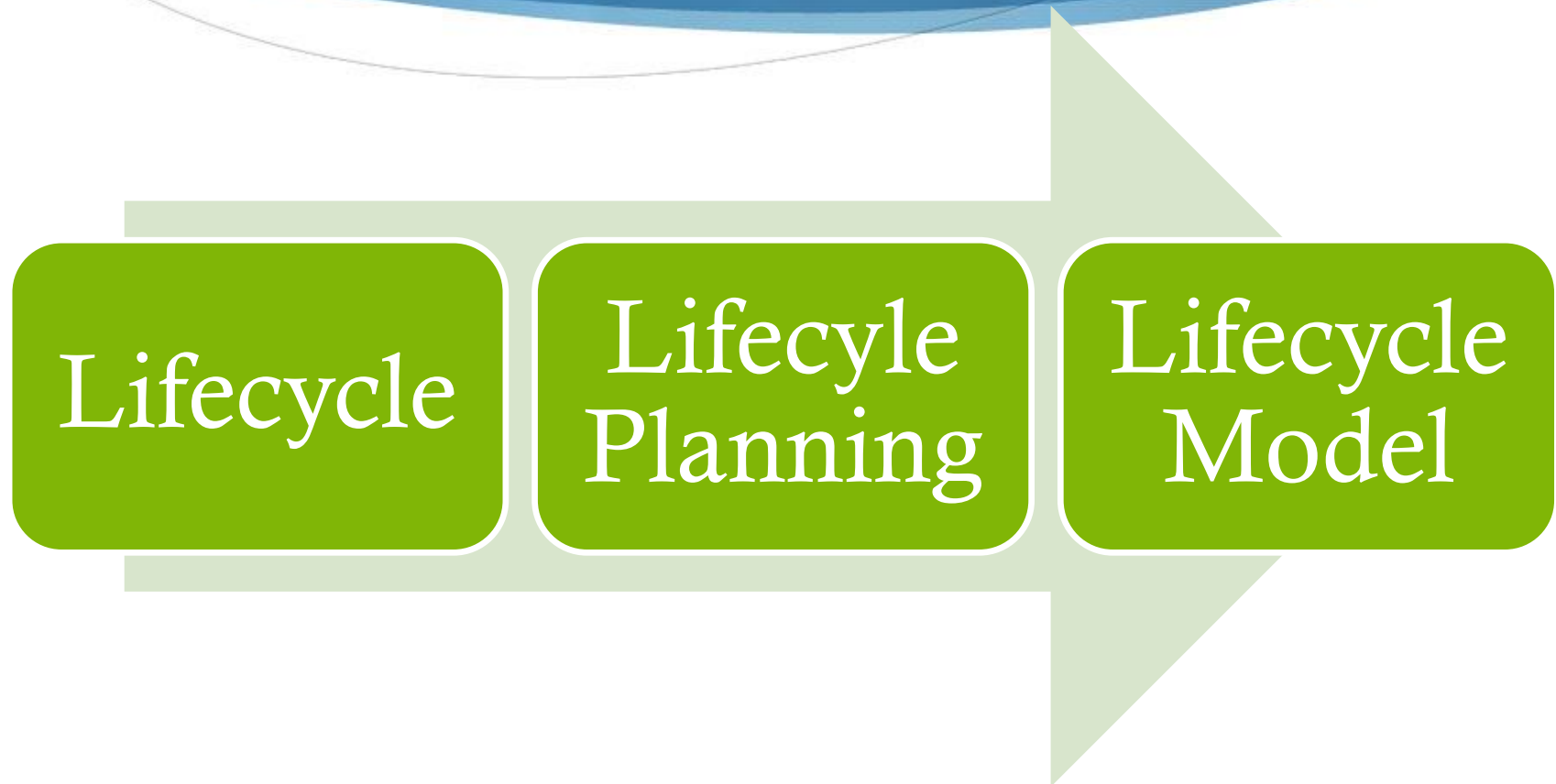
- ◆ Attendance
- ◆ Review Life Cycle Planning (McConnell – Chapter 7)
- ◆ Estimation & Scheduling (McConnell – Chapters 8 & 9)
- ◆ Assignment 2 details
- ◆ Midterm (P)review
- ◆ The next few weeks

Review

Lifecycle Planning
- McConnell, Chapter 7



Lifecycle – Planning – Model



Lifecycle Planning

Different projects have different development needs – even if they all need to be developed as soon as possible

Requirements Considerations

How well do I understand the system architecture?

Am I likely to need to make major architectural changes midway through the project?

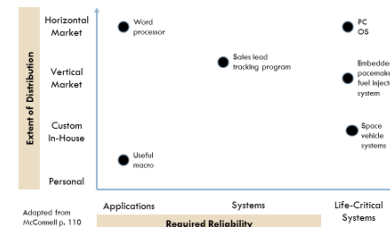
Architectural Considerations

How well do my customer and I *both* understand the requirements at the beginning of the project?

How likely are requirements to change?

Reliability Considerations

How much reliability do I need?



Future Version Considerations

How much do I need to plan ahead and design ahead during this project for future versions?

More Lifecycle Models

Some to Explore

- Spiral
- Modified Waterfall – 3x
- Evolutionary Prototyping
- Design-to-Schedule
- Staged Delivery
- Evolutionary Delivery, including Incremental Development Practices

Some others

- Code-and-Fix
- Design-to-Tools
- COTS
- Scrum
- XP

Lifecycle Strengths and Weaknesses

Table 7-1. Lifecycle Model Strengths and Weaknesses

Lifecycle Model Capability	Pure Waterfall	Code-and-Fix	Spiral	Modified Waterfalls	Evolutionary Prototyping
Works with poorly understood requirements	Poor	Poor	Excellent	Fair to excellent	Excellent
Works with poorly understood architecture	Poor	Poor	Excellent	Fair to excellent	Poor to fair
Produces highly reliable system	Excellent	Poor	Excellent	Excellent	Fair
Produces system with large growth envelope	Excellent	Poor to fair	Excellent	Excellent	Excellent
Manages risks	Poor	Poor	Excellent	Fair	Fair
Can be constrained to a predefined schedule	Fair	Poor	Fair	Fair	Poor
Has low overhead	Poor	Excellent	Fair	Excellent	Fair
Allows for midcourse corrections	Poor	Poor to excellent	Fair	Fair	Excellent
Provides customer with progress visibility	Poor	Fair	Excellent	Fair	Excellent
Provides management with progress visibility	Fair	Poor	Excellent	Fair to excellent	Fair
Requires little manager or developer sophistication	Fair	Excellent	Poor	Poor to fair	Poor

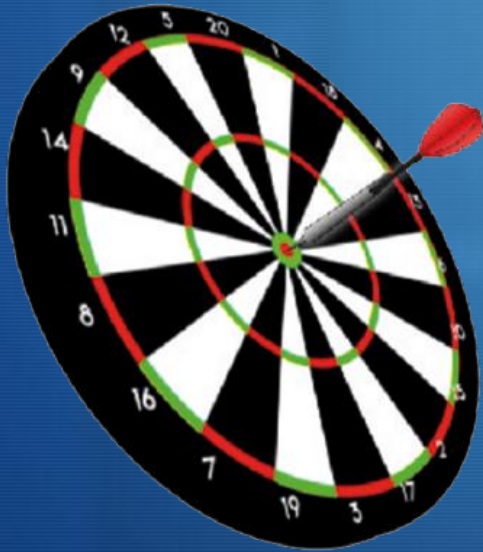
Estimation

Chapter 8



Estimation,

questions to start



- What is an “estimate”?
- How accurate are estimates of software development project effort and duration?
- What makes estimation so difficult?
- What do people want when they ask for an estimate?
- Who should do the estimating?
- What kind of estimation technique(s) have you used?

Intro to Estimating

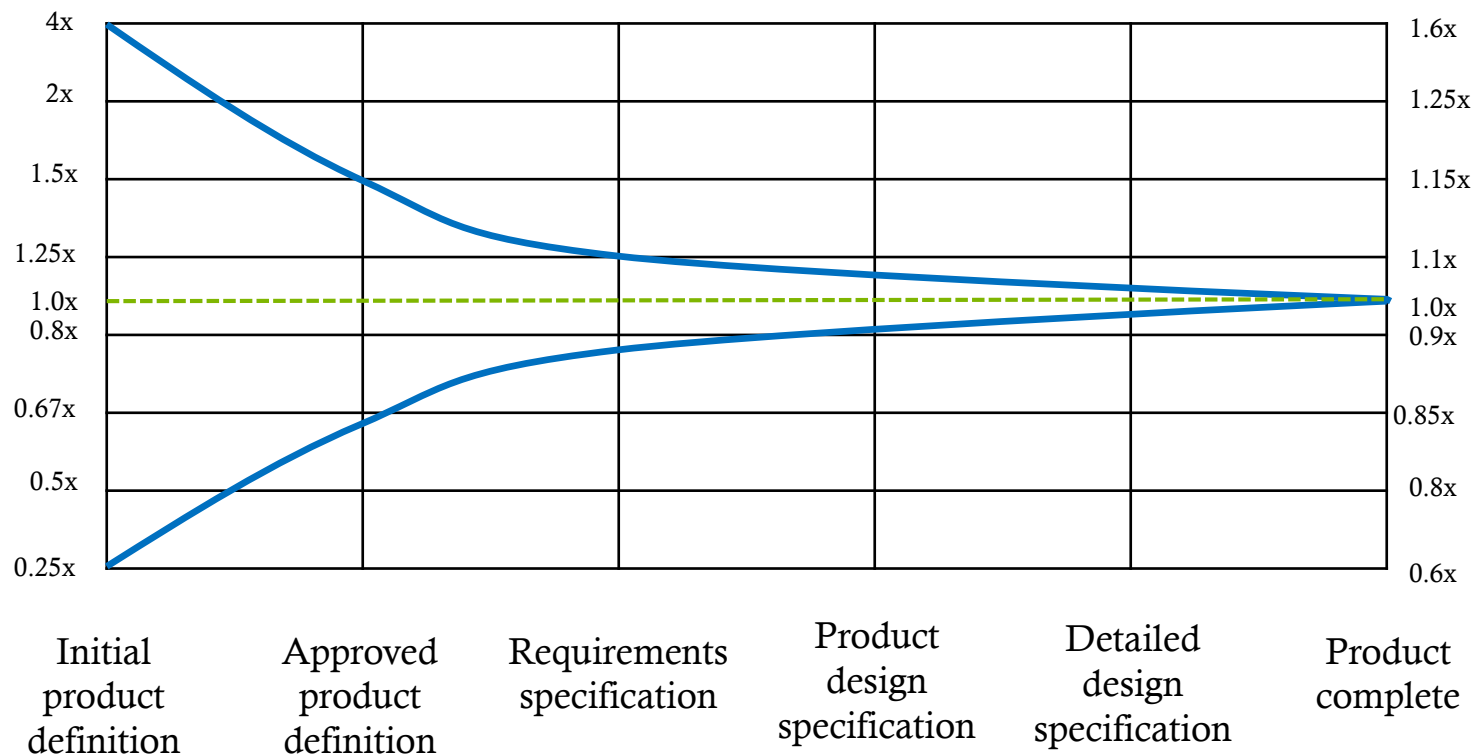
- ◆ Goal should not be to come in too high or low, but right on
- ◆ Estimation takes time
 - ◆ Olympic 2010 bid: 10 page document, \$100 million
- ◆ Accuracy vs. Precision
- ◆ Give me a software estimate (what units is it in?)

Estimate-Convergence Graph

Variability in the estimate
of Project Scope
(effort, cost, features)

The “Cone of Uncertainty”
Based on common project milestones

Project Schedule

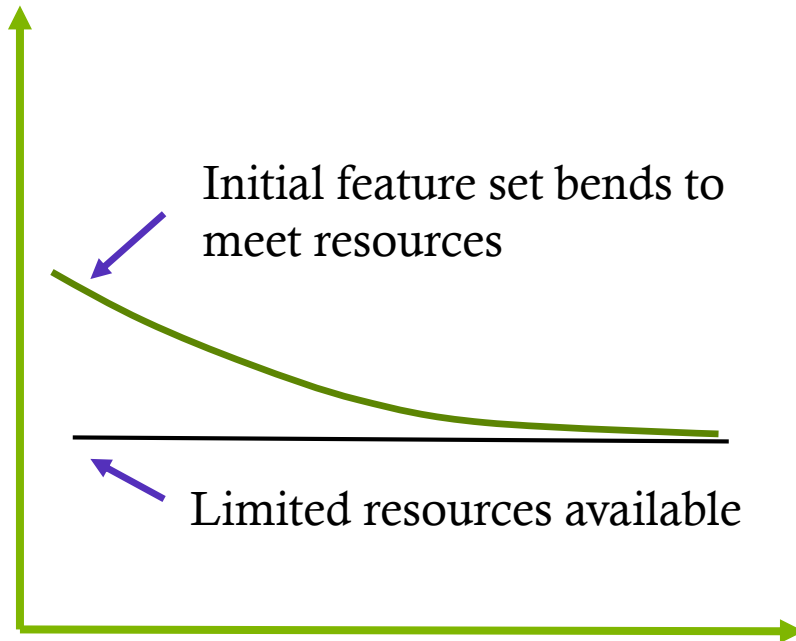


What is your experience with the rules that this graph is trying to illustrate?

Estimation vs. Control

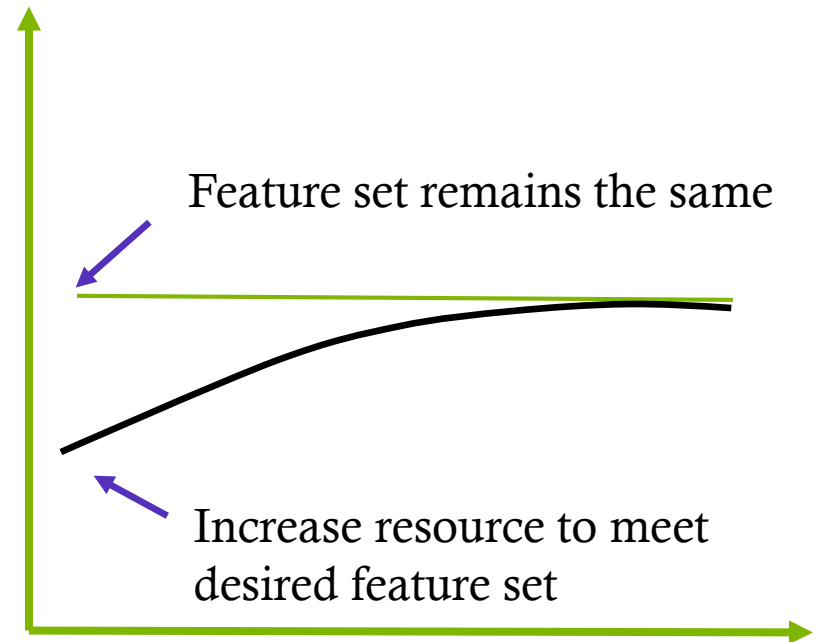
Which project would you prefer to work on?

Product Size



Evolution
of Project

Product Size



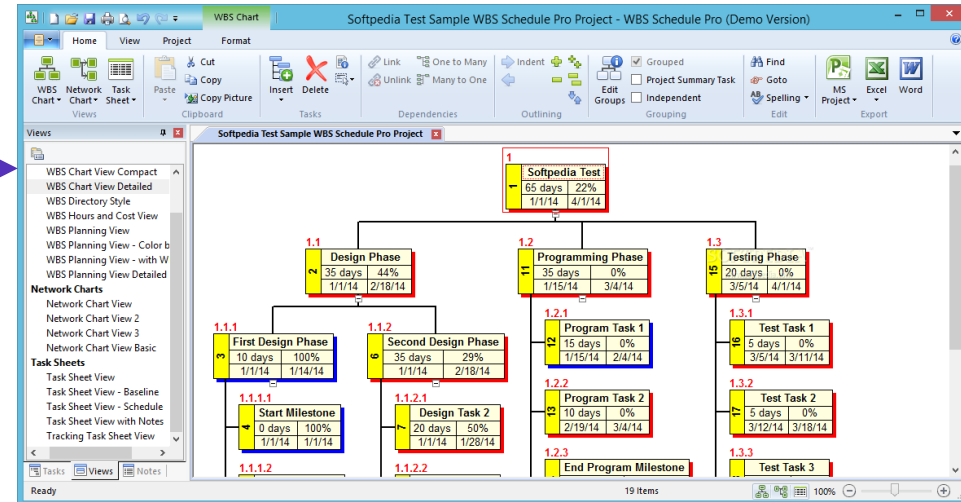
Evolution
of Project

Methods of Estimation

Historical Data

Bottom up approach (WBS) →

- Break down work into smaller tasks
- Estimate effort of tasks
- Sum them upwards



Function points (McConnell pg 174)

- Based in inputs, outputs, inquiries, internal/external interfaces
- Calculate function points, then compare to historical work

3-point or PERT

- $$[\text{Optimistic} + (4 * \text{Most Likely}) + \text{Pessimistic}] / 6$$

Methods of Estimation

- ◆ Delphi

- ◆ Anonymous polling of large groups of experts

- ◆ Scrum: story points

- ◆ What is a “point” worth?

It's all relative, but the system must be consistent

Example of story point estimation:

- ◆ hand everyone cards (0, 1, 2, 3, 5, 8, 13, 20)
 - ◆ display at the same time (no influence)
- ◆ discuss high/low estimates
- ◆ come to a consensus, timebox (X minutes per story estimate)

Estimation Tips

- ◆ Avoid “off-the-cuff” estimates
 - ◆ Allow time/plan for estimate
 - ◆ Use previous project data
 - ◆ Use developer-based estimates
 - ◆ Estimate by walk-through
 - ◆ Estimate by categories
 - ◆ Estimate at low level of detail
 - ◆ Don’t omit common tasks
 - ◆ Use several techniques and compare results
 - ◆ Recalibrate
- Discuss how Agile plays a role in recalibrating estimates

Estimation Presentation Styles

- ◆ Plus-or-minus qualifiers: e.g. 3 months +/- 3 weeks
- ◆ Ranges, e.g. 4 - 5 months
- ◆ Identify/quantify risks
- ◆ Best/Worst/Likely Cases
- ◆ Coarse dates and time periods, not specific dates (e.g. Q3)
- ◆ Confidence factors with probabilities

Estimation Case Study

Case Study 8.2

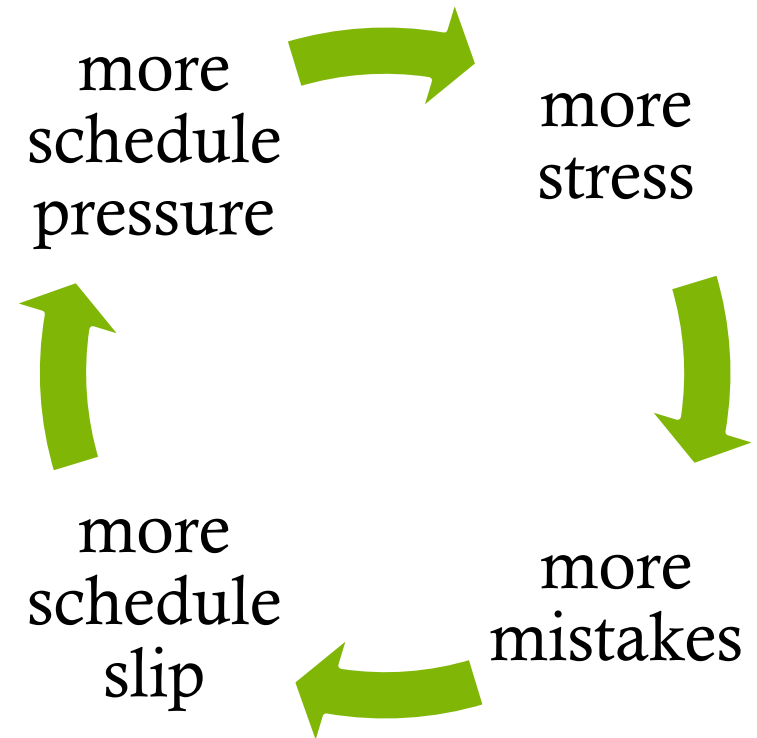
Careful Project Estimation

pp. 200-202

- Review Case Study 8.2
 - Why does “management” want specifics?
 - Which steps are completed to get the estimate down from quarters to months and then to weeks of effort estimates?
 - How does George manage expectations?
 - What important question does management not ask George at the initial meeting?
- Review on your own, then pair up

Scheduling

Chapter 9



A summary, before we even begin

- ♦ Ideally, scheduling should be a simple procedure, flowing from good estimates
- ♦ We've already discussed a lot about schedules
 - ♦ Classic mistakes (top 10 schedule risks)
 - ♦ Risk Management (reducing schedule risk)
 - ♦ Core Issues (distribution and reliability, quality vs. schedule)
 - ♦ Lifecycle Model
- ♦ Two different things: Estimates vs. Schedule target

Schedule Risk Identification

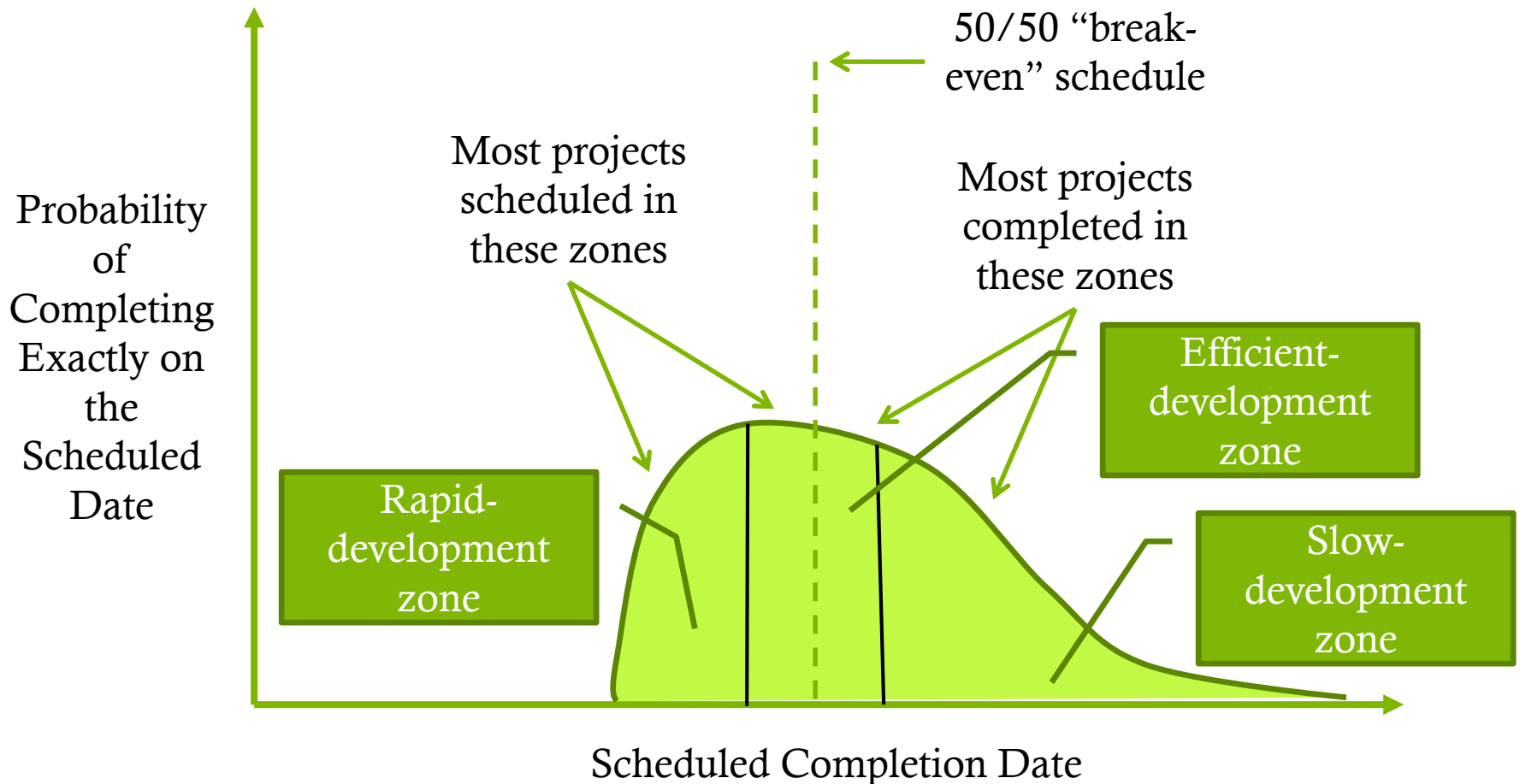
Remember this?

People-Related	Process-Related	Product-Related	Technology-Related
1. Undermined motivation 2. Weak personnel 3. Uncontrolled problem employees 4. Heroics 5. Adding people to a late project 6. Noisy, crowded offices 7. Friction between developers and customers 8. Unrealistic expectations 9. Lack of effective project sponsorship 10. Lack of stakeholder buy-in 11. Lack of user input 12. Politics placed over substance 13. Wishful thinking	14. Overly optimistic schedules 15. Insufficient risk management 16. Contractor failure 17. Insufficient planning 18. Abandonment of planning under pressure 19. Wasted time during the fuzzy front end 20. Shortchanged upstream activities 21. Inadequate design 22. Shortchanged quality assurance 23. Insufficient management controls 24. Premature or too frequent convergence 25. Omitting necessary tasks from estimates 26. Planning to catch up later 27. Code-like-hell programming	28. Requirements gold-plating 29. Feature creep 30. Developer gold-plating 31. Push me, pull me negotiation 32. Research-oriented development	33. Silver-bullet syndrome 34. Overestimated savings from new tools or methods 35. Switching tools in the middle of a project 36. Lack of automated source

What happens if a quality product is delivered late?

Expectations of Scheduling

Remember this?



Rapid Development Look-Alikes

Remember this?

- 💧 Runaway Prevention
- 💧 Predictability
- 💧 Lowest Cost
- 💧 Fixed Drop-Dead Date
- 💧 Unpaid Overtime

Overly Optimistic Schedules

- ◆ External immovable deadline (tradeshow, Christmas, regulations)
- ◆ Sales people under estimating (trying to make the deal)
- ◆ Upper management choosing “best case scenario” rather than a range
- ◆ Manager ineffective during customer negotiations
- ◆ Project Manager believes team will work harder with tighter deadlines
- ◆ Developers under estimating (too confident of skills)

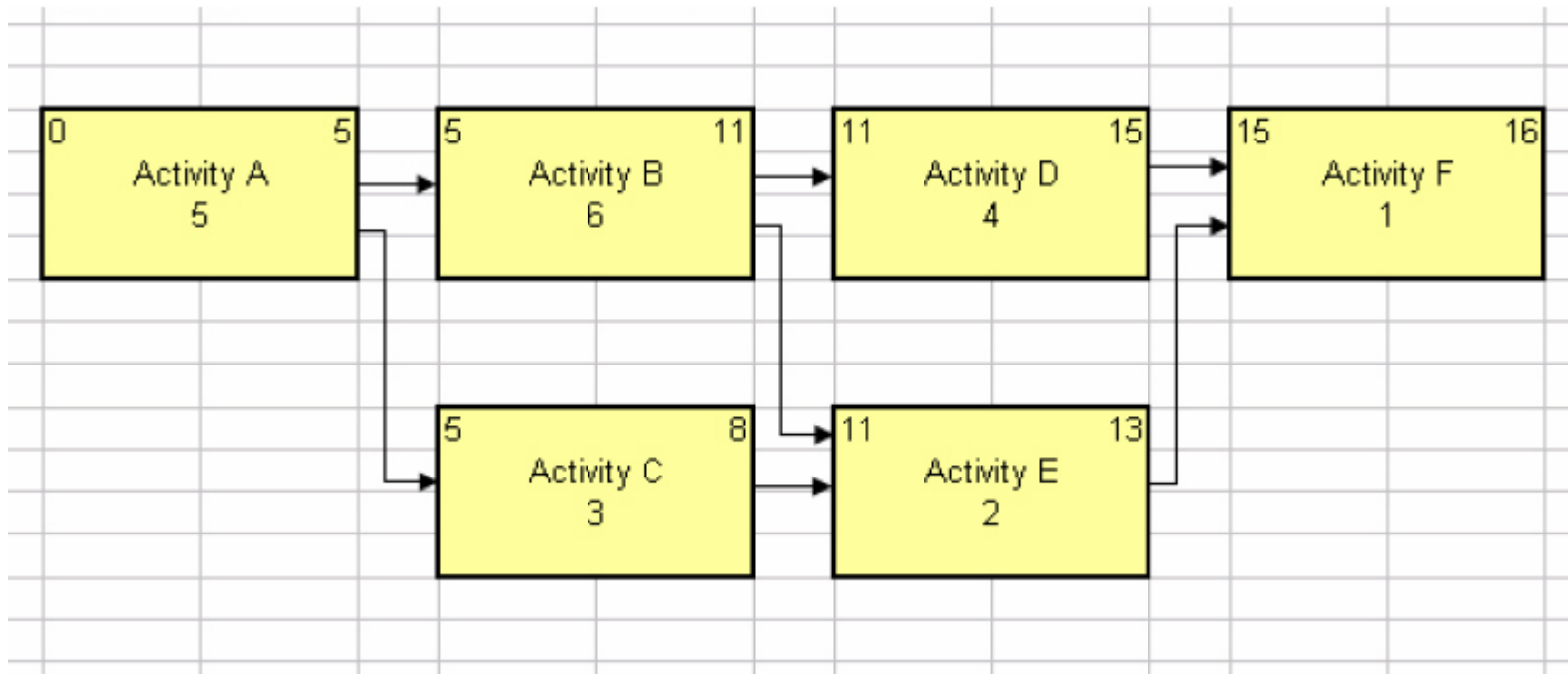
Negotiating - “Getting to Yes”

◆ Principled Negotiation

1. Separate the **people** from the problem
2. Focus on **interests**, not positions
3. Invent **options** for mutual gain
4. Insist on using **objective criteria**

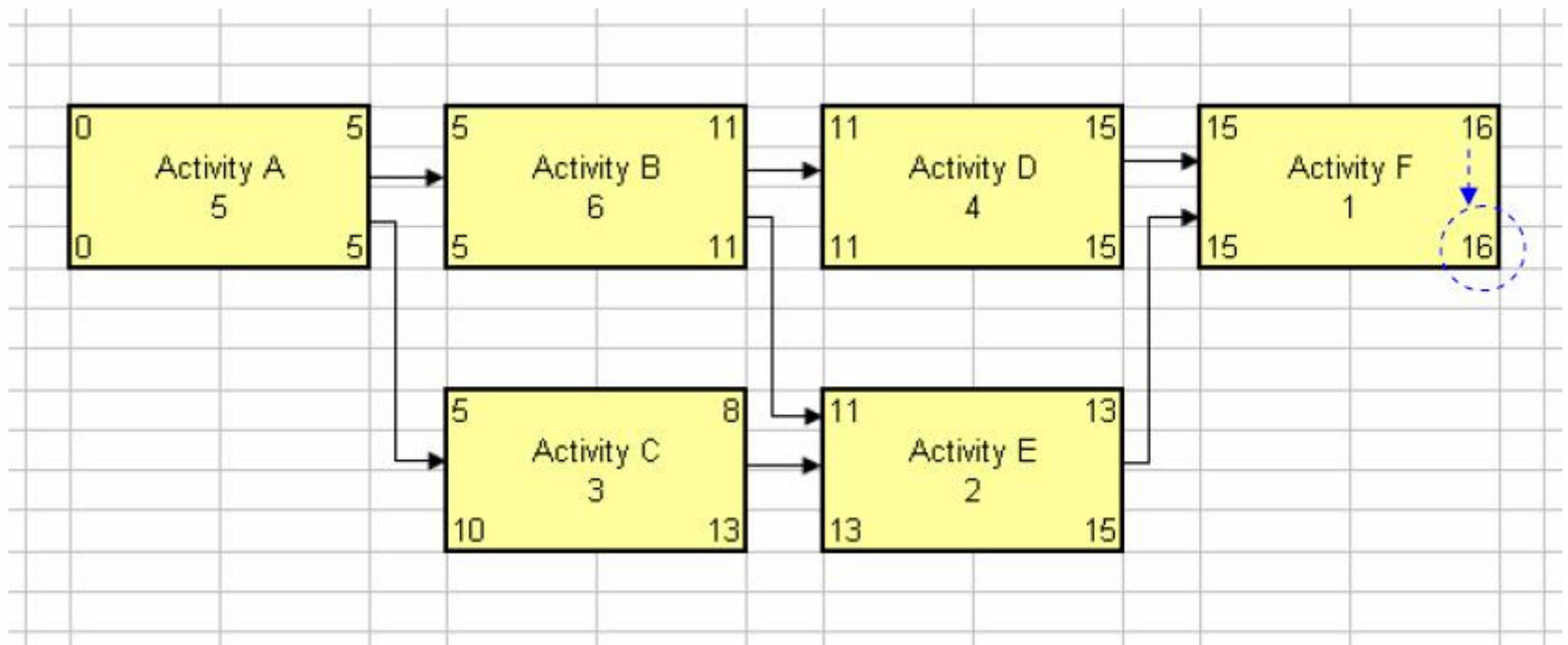
Overview: Critical Path Method

- ◆ Critical Path,
Critical Path Activity
- ◆ Predecessor / Successor
- ◆ Forward pass:
Early Start/Finish
- ◆ Duration, Lag



Overview: Critical Path Method

- Backward pass: Late Start/Finish
- Float, Total Float



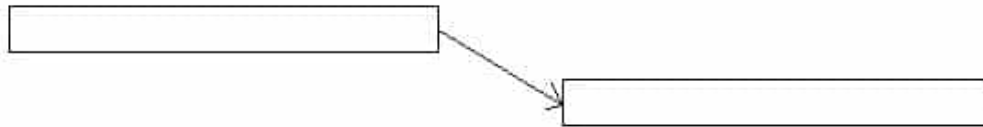
Schedule Compression

- ◆ PMI says:
 - ◆ Fast Tracking
 - ◆ Crashing
- ◆ BUT at some point, that's it,
your schedule cannot be compressed any further

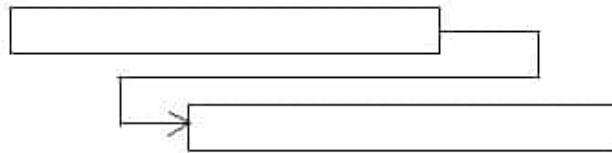
Schedule Compression: Fast Tracking

- Review Critical Path activities to see which can be done in parallel, or partially parallel
 - Does not increase cost
 - Does increase risk

Before Fast Tracking



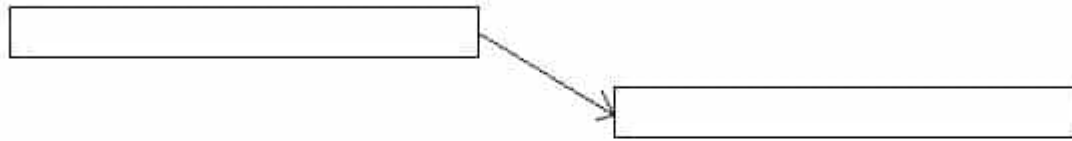
After Fast Tracking



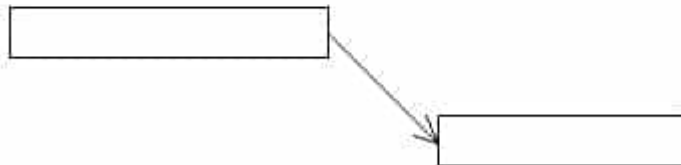
Schedule Compression: Crashing

- ◆ Add extra resources to shorten Critical Path activities
 - ◆ Does increase cost, but not as much risk as Fast Tracking
 - ◆ Cost increases at a faster rate than the reduction of schedule

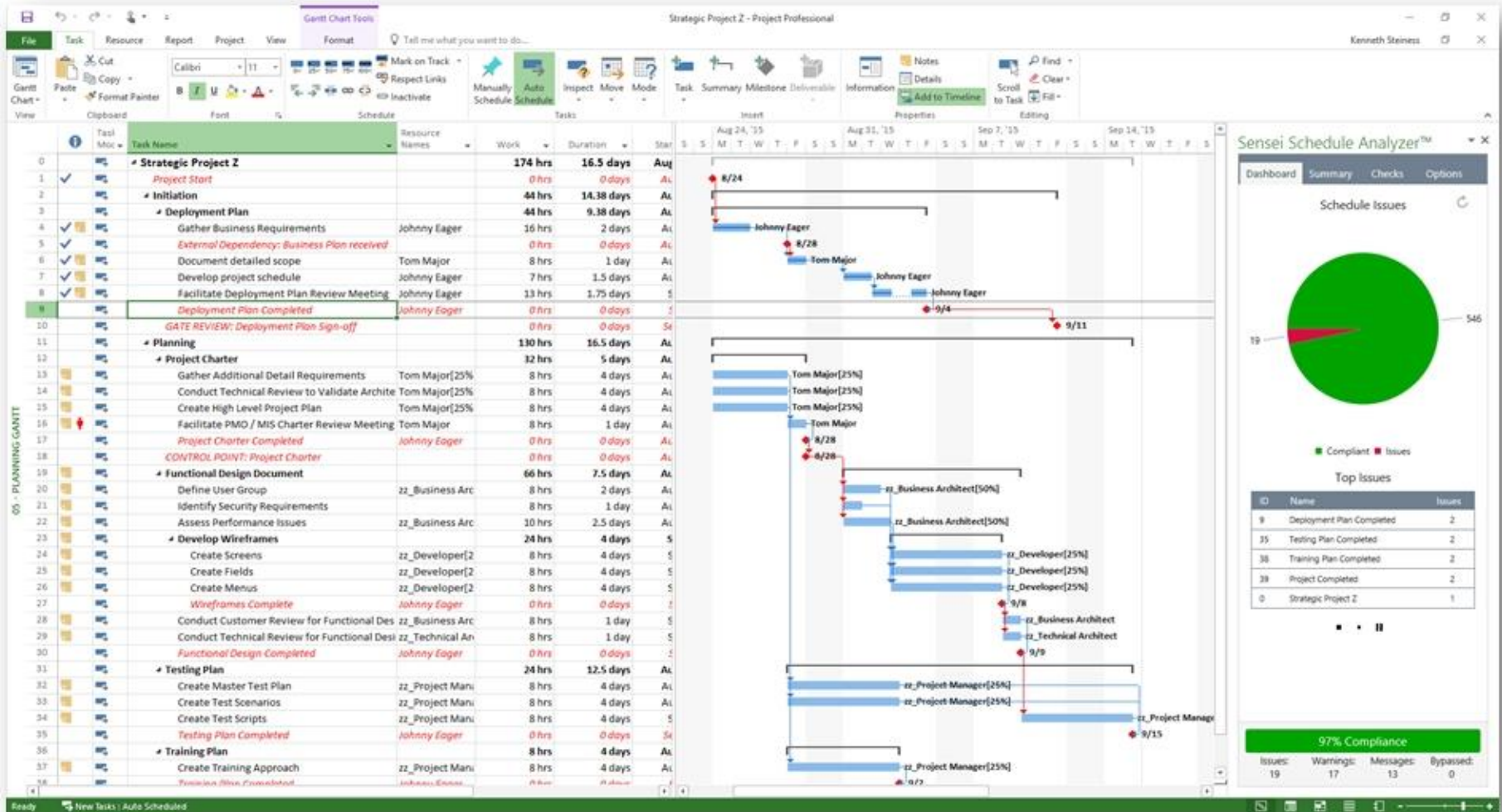
Before crashing



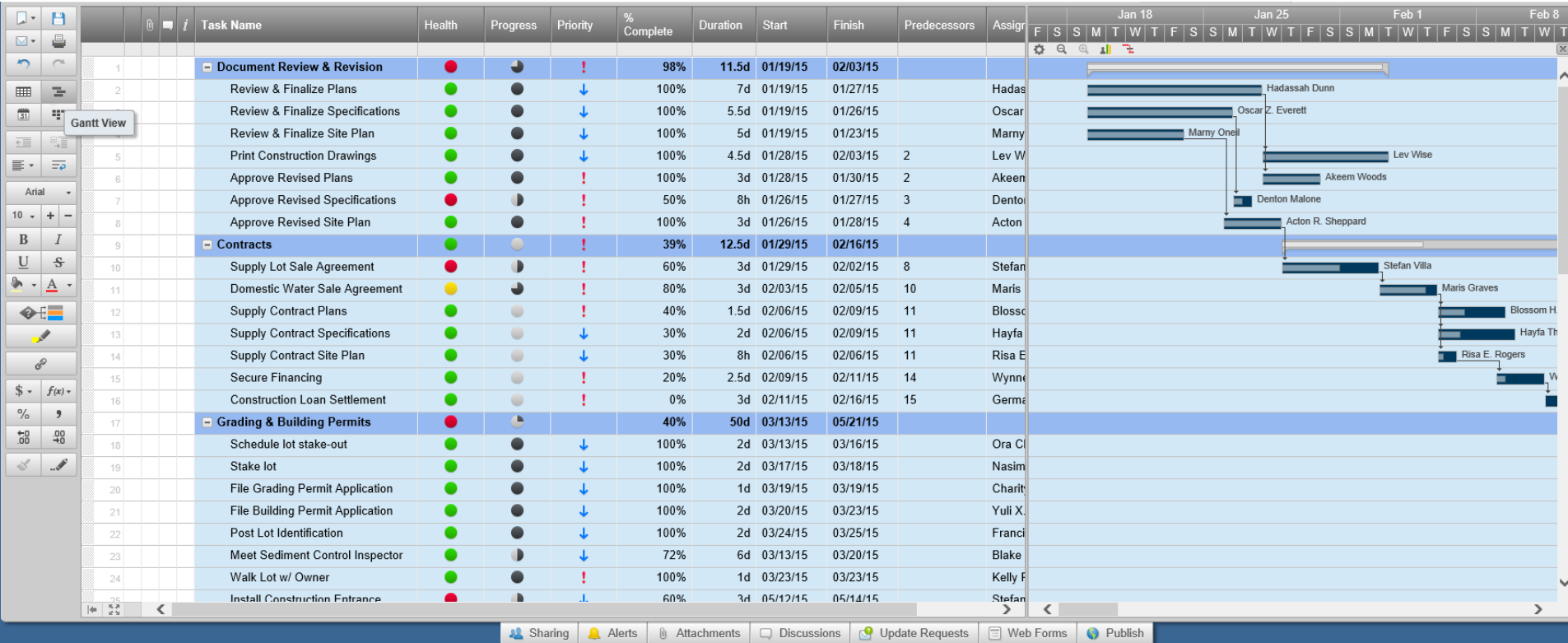
After crashing



Schedule example – MS Project



Schedule example – Smart Sheet



Schedule example – Excel (Donna)

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1			Month	July				August				September				October				November						
2			Week of	4	11	18	25	1	8	15	22	29	5	12	19	26	3	10	17	24	31	7	14	21	28	
3	Main Task	LOE	Sub-Task																							
4						Req's	UX	architecture	dev, revisions, bug fixes, QA												wrap up					
6	Requirements		finalize																							
7			sign-off			29																				
8	UX Design		7 design creation																							
9			2 review and revisions																							
10			sign-off					12																		
11	Architecture		5 design																							
12			3 review and revisions																							
13			design sign-off									26														
14			security sign-off									26														
15	Development		28 Dev1																							
16	(assuming 2 devs)		28 Dev2																							
17			8 UI																							
18			5 Settings page																							
19			4 Active Directory service connector																							
20			15 Product configuration entitlement mapping																							
21			8 integration with user management API																							
22			5 synchronization service																							
23			5 audit trail																							
24			customer code review														3									
25			security review														3									
26			revisions																							
27			code sign-off																							
28			security sign-off																							
29	QA		10 test case design																							
30			10 test case execution																							
31	Dev & QA		5 bug fixing & verification																							
32	Documentation		5 during development																							
33			5 post-development																							
34	Completion		1 delivery																							
35			2 knowledge transfer																							
36			2 deployment support																							

Assignment 2

- ◆ Form teams of 2 or 3, self-enroll into D2L group
- ◆ Wait for my D2L News item, confirming groups
- ◆ Submit presentation PDF to D2L
 - ◆ **Wednesday Marcy 7th 11:59pm for ALL TEAMS**
 - ◆ For fairness, same deadline, regardless of presentation date
- ◆ 10 minute presentation, March 8th, 15th, or 22nd

Midterm Exam

- ◆ Hand-written, no external aids (books, cheat sheets, etc.)
- ◆ Expository, longer form answers
- ◆ Application of the concepts in case study scenarios
- ◆ Some recall, but insight is the most important

Midterm Exam

- ◆ Estimation and Scheduling
 - ◆ Cone of Uncertainty
- ◆ Core Issues considerations
 - ◆ Value line and “carefulness”
- ◆ Lifecycle Model selection considerations
 - ◆ Focus on **Spiral Development & Evolutionary Delivery** lifecycle models
- ◆ Classic Mistakes
 - ◆ Esp. the 10 most common schedule risks
- ◆ Four Pillars and Four Dimensions of Dev Speed
- ◆ Development Fundamentals
 - ◆ PM Iron Triangle
- ◆ Risk Management categories, process and maturity levels

The next few weeks

- ◆ Next week: Midterm Exam in class
- ◆ Week 8
 - ◆ Customer-Oriented Development (McConnell Chapter 10)
- ◆ Week 9 & 10
 - ◆ Break from McConnell
 - ◆ Meyers-Briggs Type Indicators (MBTI)
- ◆ Weeks 9, 10, 11
 - ◆ Assignment 2 presentations

Class Exercise

On D2L



Comp8081

end of Week 6

Donna Turner

