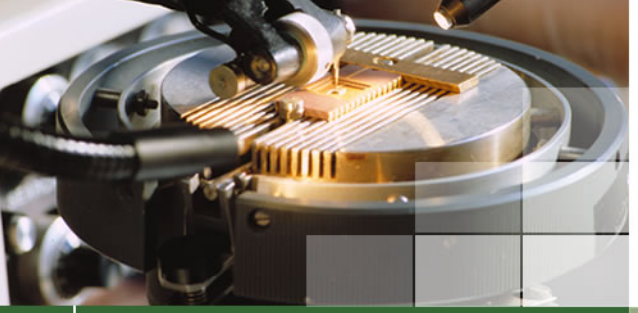


POLITECNICO DI MILANO

Dipartimento di
Elettronica e Informazione

Planning and Managing Software Projects 2013-14

Class 16

People Dimension

Emanuele Della Valle

http://emanueledellavalle.org

Credits

- This slides are largely based on Prof. John Musser class notes on “Principles of Software Project Management”
- Original slides are available at http://www.projectreference.com/
- Reuse and republish permission was granted

Today

- Project Roles
- Staffing profile
- Hiring
- Team models and successful projects
- Optimal team size
- Tools: RAM and Skill Matrix

Project Roles

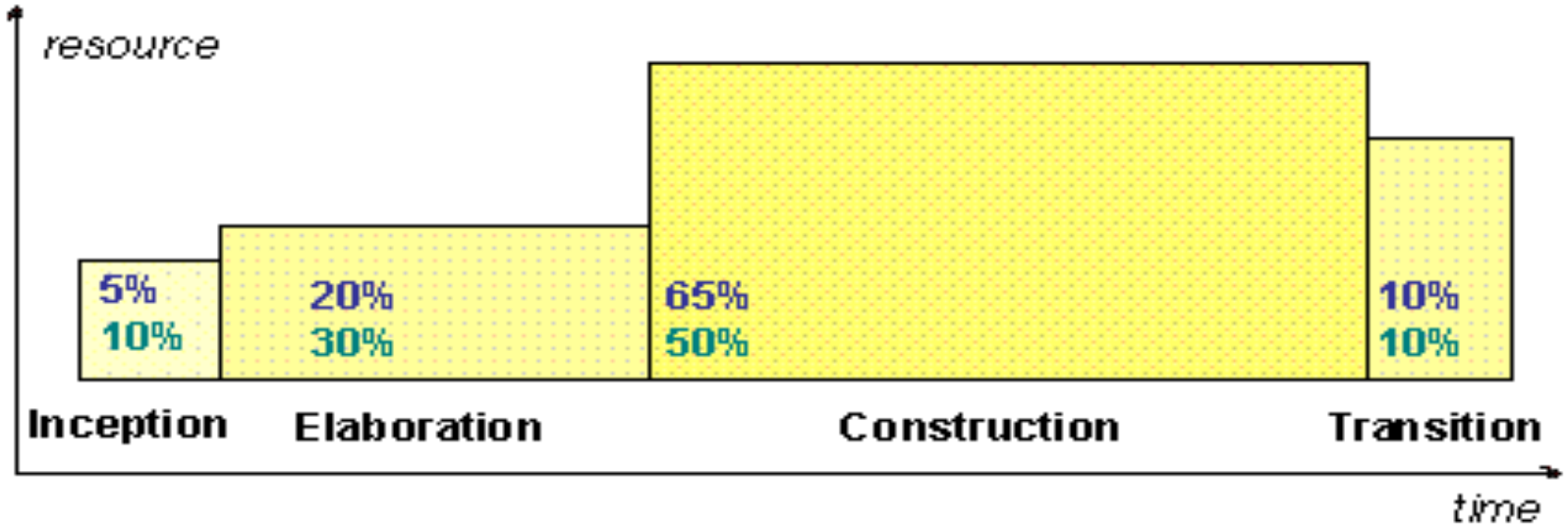
- Programmers (system engineers)
  - Technical lead, architect, programmer, Sr. programmer
- Quality Assurance (QA) engineers (testers)
  - QA Manager, QA Lead, QA staff
- DBAs
  - DB Administrator, DB Programmer, DB Modeler
- CM engineers (build engineers)
- Network engineers, System Administrators
- Analysts (business analysts)
- UI Designers
- Information Architects
- Documentation writers (editors, documentation specialist)
- Project manager
- Other
  - Security specialist, consultants, trainer

Project Roles & Homework 4

- You need to decide which of these are necessary for your class project
- Depends on what you're building
  - How big is it?
  - Is it UI intensive? Data intensive?
  - Are you installing/managing hardware?
  - Do you need to run an operations center?
  - Is it in-house, contract, Commercial off-the-shelf (COTS), etc?
- Depends on your budget
- More about it in class 15 in the computer laboratory

Staffing Profile

- Projects do not typically have a ‘static team size’
- Who and how many varies as needed



Legend:
Actual Effort (% of project total)
Schedule (% of project total)

Roll-on & Roll-off

- PM must have a plan as to how & when
- Roll-on
  - Hiring or ‘reserving’ resources
  - Ramp-up time
    - Learning project or company
- Roll-off
  - Knowledge transfer
  - Documentation
  - Cleanup

Staffing Management Plan

- Part of Software Development Plan
- Includes
  - What roles needed, how many, when, who
  - Resource assignments
  - Timing: start/stop dates
  - Cost/salary targets (if hiring)
- Project Directory
  - Simply a list of those involved with contact info.
- Team size: often dictated by budget as often as any other factor

Hiring

- “Hire for Attitude, Train for Skill”
- Look for: “Smart, Gets Things Done”
- Balance the team

Optional Reading

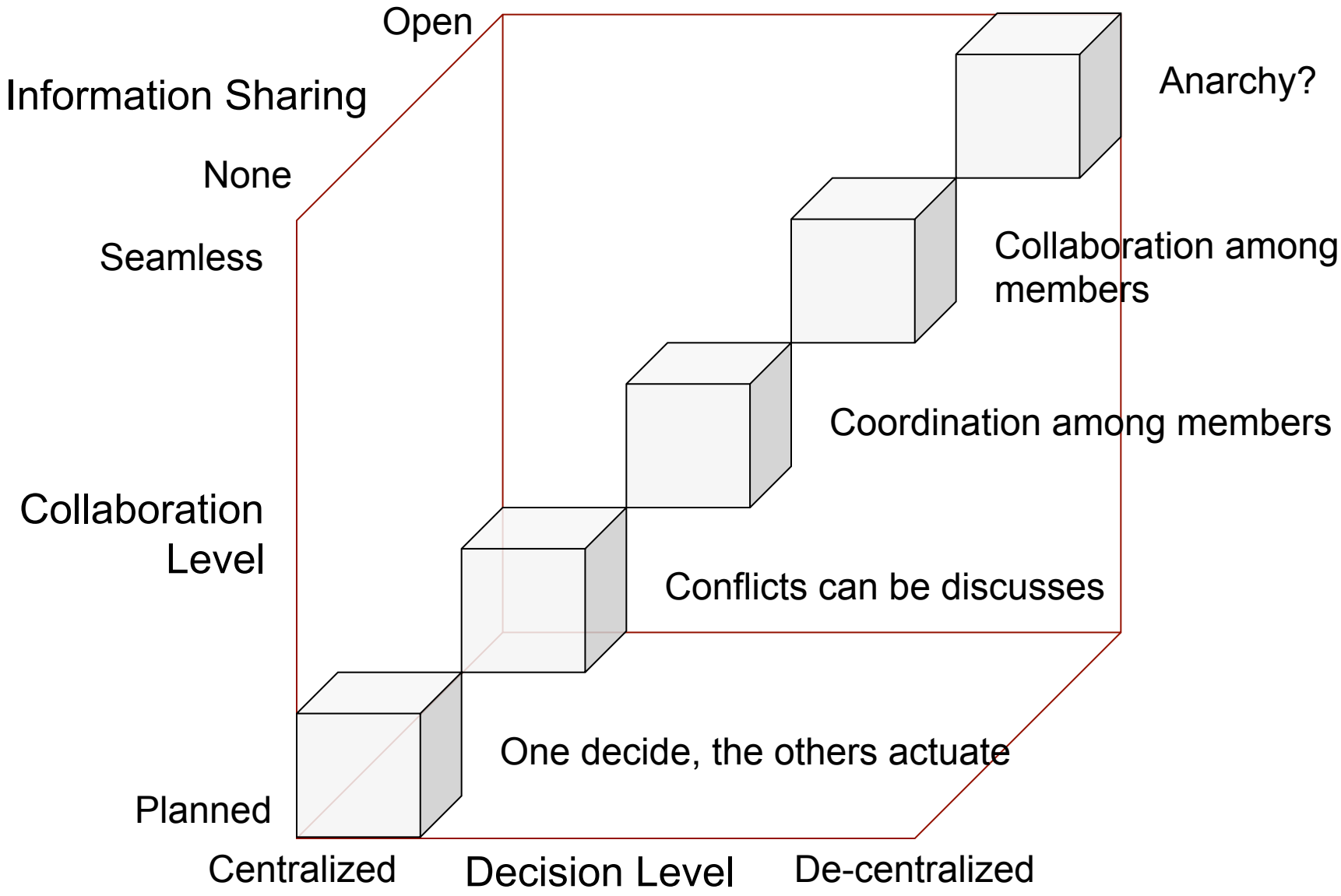
- joelonsoftware's "Guerilla Guide to Interviewing"
  - http://www.joelonsoftware.com/articles/fog0000000073.html
  - http://www.joelonsoftware.com/articles/GuerrillaInterviewing3.html

Team Models

- 1st: What's the team's objective?
  - Problem resolution
    - Complex, poorly-defined problem
    - Focuses on 1-3 specific issues
      - Ex: fixing a showstopper defect
    - Sense of urgency
  - Creativity
    - New product development
  - Tactical execution
    - Carrying-out well-defined plan
    - Focused tasks and clear roles

No single team model is best for all projects

Decision vs. Collaboration vs. Information Sharing



Be Aware of Potential Conflicts

[source http://www.globalnerdy.com/2011/08/03/the-war-between-developers-designers-and-project-managers/]

Business Team

- Most common model
- Technical lead + team (rest team at equal status)
- Hierarchical with one principal contact
- Adaptable and general
- Variation: Democratic Team
  - All decisions made by whole team
  - See Weinberg's "egoless programming" model [1]
    - http://www.ingeniumweb.com/blog/post/the-ten-commandments-of-egoless-programming/1023/

[1] Gerald M. Weinberg, "Egoless Programming," IEEE Software, vol. 16, no. 1, pp. 118-120, Jan./Feb. 1999, doi:10.1109/MS.1999.744582

Chief-Programmer Team

- From IBM in 70's
  - See Brooks and Mythical Man-Month
- a.k.a. 'surgical team'
- Puts a superstar at the top
  - Others then specialize around him/her
    - Backup Programmer
    - Co-pilot or alter-ego
    - Administrator
    - Toolsmith
    - "Language lawyer"
- Issues
  - Difficult to achieve
  - Ego issues: superstar and/or team
- Can be appropriate for creative projects or tactical execution

"Skunkworks" Team [1]

- Put a bunch of talented, creative developers away from the mother ship
  - Off-site literally or figuratively
- Pro: Creates high ownership & buy-in
- Con: Little visibility into team progress
- Applicable: exploratory projects needing creativity
  - Not on well-defined or narrow problem

[1] http://searchcio.techtarget.com/definition/skunkworks

SWAT Team [1]

- Highly skilled team
- Skills tightly match goal
- Members often work together
- Ex: security swat team
- The team model for tactical execution

[1] http://en.wikipedia.org/wiki/SWAT

Other (bad) kind of teams

Check out http://www.scottberkun.com/blog/2007/asshole-driven-development/ for

- Asshole Driven development
- Cognitive Dissonance development
- Cover Your Ass Engineering
- Development By Denial
- Get Me Promoted Methodology

Wrapping up

Team Model	Problem Resolution	Creativity	Tactical Execution
Business Team	***	*	**
Chief-Programmer Team		***	**
"Skunkworks" Team		***	
SWAT Team			***

LEGEND

*** Best suited

* Can be used

The myth of additional manpower

- The problem
  - "Adding manpower to a late project makes it later"
    - Brooks Law http://en.wikipedia.org/wiki/Brooks%27s_law
- The fix
  - Remember!
    - Q: "How does a project get to be a year late?"
    - A: "One day at a time!"
  - Consider the 50% not-coding time when planning
  - Define clearly measurable milestones
    - No "fuzzy" milestones
  - Reduce the role of conflict among persons
  - Identify the "true status" of a task
    - It's impressive how much effort is needed to move a 90% done task to a 100% done task
  - Don't add manpower to a late project, reschedule it!
    - More in class 19 in "Project Recovery" section

Large teams

- Communication increases multiplicatively
  - Square of the number of people
  - 50 programmers = 1200 possible paths
  - Communication must be formalized
    - e.g. use deliverables
- Always use a hierarchy
- Reduce units to optimal team sizes

Team Size

- What is the optimal team size?
  - 4-6 developers
    - Tech lead + developers
  - Small projects inspire stronger identification
  - Increases cohesiveness
  - QA, operations, and design on top of this
  - Always less then 10!

Responsibility Assignment Matrix

- A resource planning tool
- Who does What
- Can be for both planning and tracking
- Identify authority, accountability, responsibility
- Who: can be individual, team or department
- Can have totals/summary at end of row or column (ex: total Contributors on a task)

Simple RAM

Item	WBS	Description		Sponsor	Developer	Developer	QA	Customer
1	1	Initiate Project		A				
2	1.1	PMP Signoff		A				R
3	1.2	Initial UI			L	C		R
4	1.3	DB Model			C	L		
5	1.4	Start Test					L	
	Legend							
	A	Approval						
	L	Lead						
	S	Secondary						
	C	Contributor						
	R	Reviewer						

Sample RAM With Stakeholders

Item		Development	Customer A	Customer B	Mgmt	QA	
Unit Test		A	S	S	R	A	
Systems Test		P	R	R	R	R	
Beta Test		P	R	R	P	R	
User Acceptance Test		A	S	S	S	S	
Accountable	A						
Participant	P						
Reviewer	R						
Sign-off Required	S						

Skills Matrix

- Another resource planning tool
- Resources on one axis, skills on other
- Skills can high level or very specific
- Cells can be X's or numeric (ex: level, # yrs.)

	Analyst	Developer (Java)	Developer (HTML)	QA Tester	Database Design
Dilbert	7	2			
Larry			8		4
Sarah	4	4			
Boss				4	
Fred					5

Questions?