Working in Teams

TEAMWORK
Large ambitious goals usually require that people work together.

1

last time: Product idea proposal!

- First assignment: Due at 9 PM, Feb 12

http://www.cs.umass.edu/~brun/class/2024Spring/CS320/productIdea.pdf

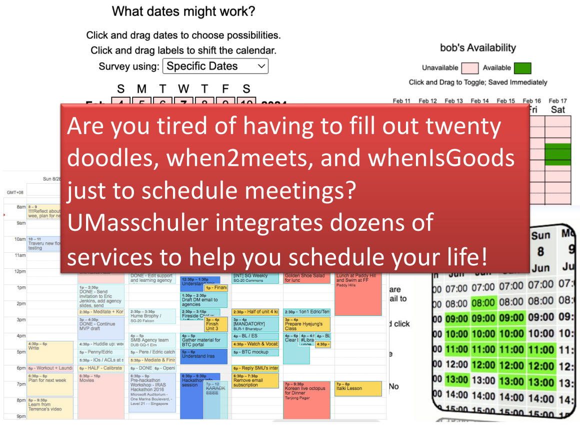
- Groups of 1 or 2
  - get into groups after class or use the Moodle class discussion forum
- Submit 4 slides:
- 3-minute presentations in class next week

2

UMasschuler

Yuriy Brun

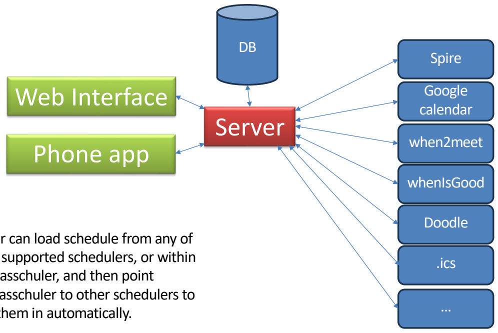
3


What dates might work?
Click and drag dates to choose possibilities.
Click and drag labels to shift the calendar.
Survey using: Specific Dates
S M T W T F S
bob's Availability
Unavailable
Available
Click and Drag to Toggle; Saved Immediately
Sun 8 Jun
07:00
08:00
09:00
10:00
11:00
12:00
13:00
14:00
15:00

Are you tired of having to fill out twenty doodles, when2meets, and whenIsGoods just to schedule meetings?
UMasschuler integrates dozens of services to help you schedule your life!

4

UMasschuler Architecture



User can load schedule from any of the supported schedulers, or within UMasschuler, and then point Umasschuler to other schedulers to fill them in automatically.

5

UMasschuler Challenges and Risks

- Schedulers may not provide API integration
  - Will need to scrape, find ways to fill out web forms automatically
- Each scheduler may require different integration methods
  - Making UMasschuler compatible with 5 or 6 schedulers will cover most users' needs
- Schedulers may change interface, breaking integration
  - Will aim to build robust integration

6

Madeline Endres' talk earlier today

What did you think?

7

Working in Teams

TEAMWORK

Large ambitious goals usually require that people work together.

8

Lecture outline

- Why is teamwork hard?
- Not getting into each other's way
- Positive teamwork

9

Team pros and cons

- Benefits
  - Attack bigger problems in a short period of time
  - Utilize the collective experience of everyone
- Risks
  - Communication and coordination issues
  - Groupthink: diffusion of responsibility; going along
  - Working by inertia; not planning ahead
  - Conflict or mistrust between team members

10

Communication: powerful but costly!

- Communication requirements increase with increasing numbers of people
- Everybody to everybody: quadratic cost
- Every attempt to communicate is a chance to miscommunicate
- But *not* communicating will *guarantee* miscommunication

11

What about conflicts?

What can cause conflicts?

- Two people want to work on the same file
  - Google docs lets you do that

But…

- What about same line?
- What about relationships between different parts of the file?
- What about design decisions?

12

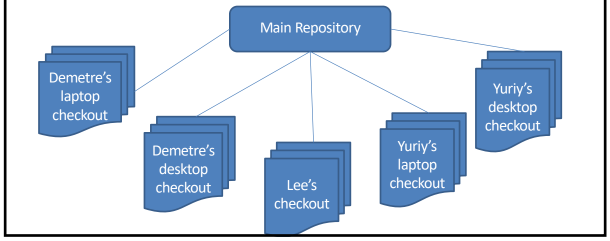
Version control

Version control aims to allow multiple people to work in parallel.

13

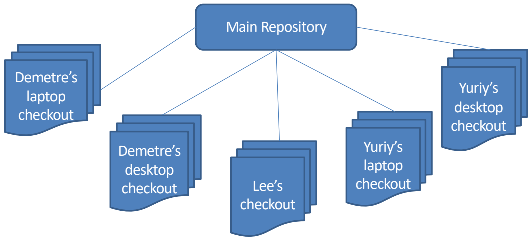
Centralized version control

- (old model)
- Examples: Concurrent Versions System (CVS)
  Subversion (SVN)



14

Doing work



- I update my checkout (working copy)
- I edit
- I update my checkout again
- I merge changes if necessary
- I commit my changes to the Main

15

Problems with centralized VC

- What if I don't have a network connection?
- What if I am implementing a big change?
- What if I want to explore project history later?

16

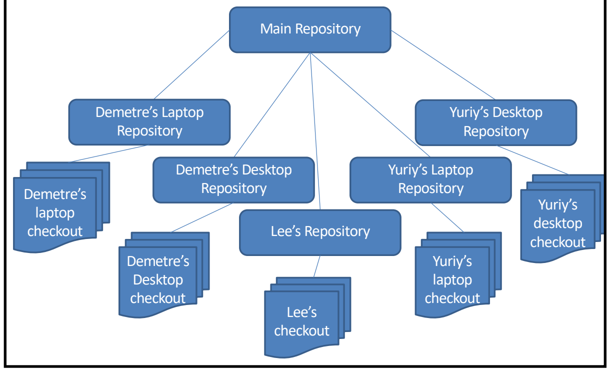
Distributed version control

(new model)

- Examples: Mercurial (Hg), Git, Bazaar, Darcs, …
- Local operations are fast (and possible)
- History is more accurate
- Merging algorithms are far better

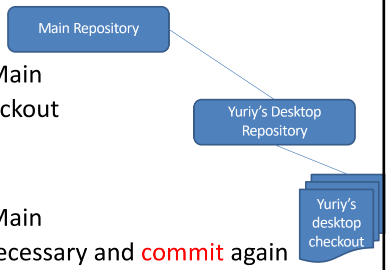
17

Distributed version control model


Main Repository
Demetre's Laptop Repository
Yuriy's Desktop Repository
Demetre's Desktop Repository
Yuriy's Laptop Repository
Lee's Repository
Demetre's laptop checkout
Demetre's Desktop checkout
Lee's checkout
Yuriy's laptop checkout
Yuriy's desktop checkout

18

Doing work

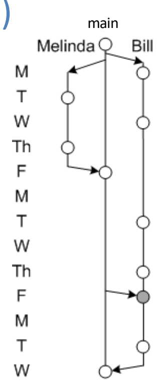


- I pull from the Main
- I update my checkout
- I edit
- I commit
- I pull from the Main
- I merge tips if necessary and commit again
- I push my changes to the Main

19

History view (log)

- Bill and Melinda work at the same time
- At the end, all repositories have the same, rich history



20

What do conflicts look like?

Crystal tool

21

What VC does the cloud provide?

- ~~code.google.com has SVN and Hg~~
- bitbucket.org has ~~Hg and~~ git
- github.com has git
- sourceforge.net has SVN, ~~CVS,~~ git, Hg, ~~Bazaar~~
- You can run whatever you want locally

22

Lecture outline

- Why is teamwork hard?
- Not getting into each other's way

➔Positive teamwork

23

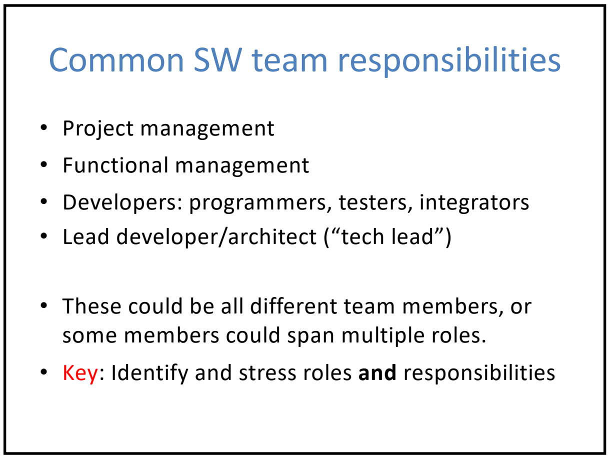
Team structures

- Tricky balance among
  - progress on the project/product
  - expertise and knowledge
  - communication needs

"A team is a set of people with complementary skills who are committed to a common purpose, performance goals, and approach for which they hold themselves mutually accountable."

– Katzenbach and Smith

24

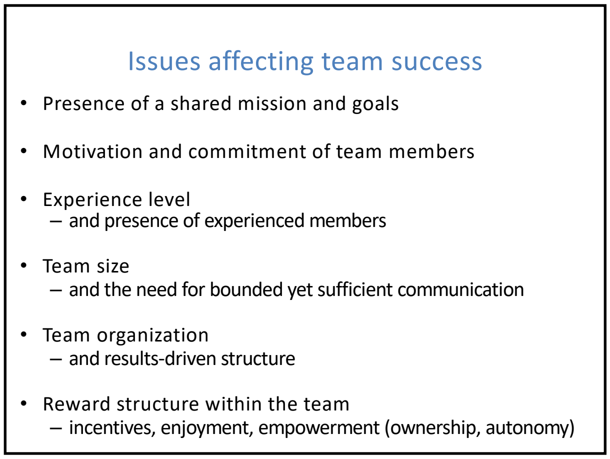


Common SW team responsibilities

- Project management
- Functional management
- Developers: programmers, testers, integrators
- Lead developer/architect (“tech lead”)

- These could be all different team members, or some members could span multiple roles.
- Key: Identify and stress roles **and** responsibilities

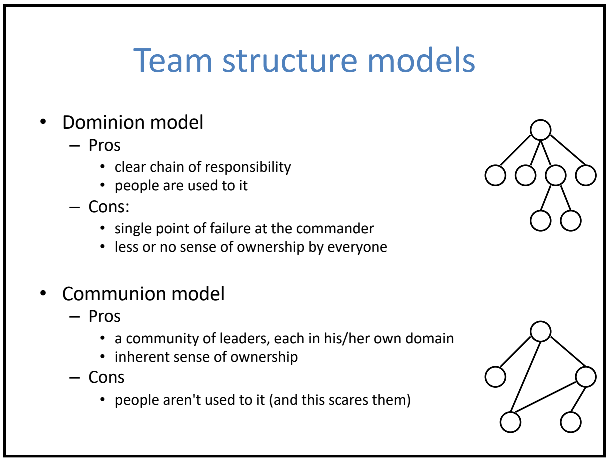
25



Issues affecting team success

- Presence of a shared mission and goals
- Motivation and commitment of team members
- Experience level
  - and presence of experienced members
- Team size
  - and the need for bounded yet sufficient communication
- Team organization
  - and results-driven structure
- Reward structure within the team
  - incentives, enjoyment, empowerment (ownership, autonomy)

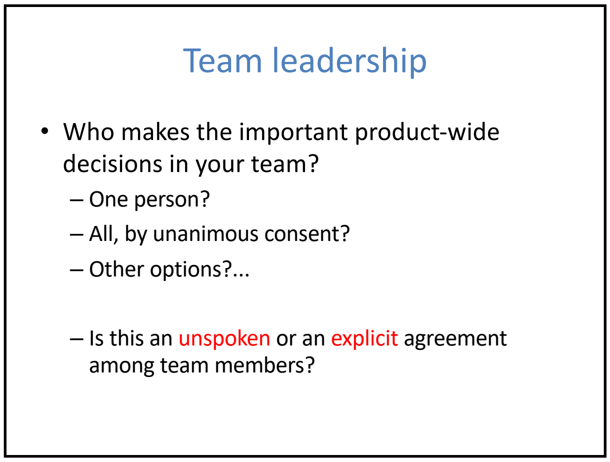
26



Team structure models

- Dominion model
  - Pros
    - clear chain of responsibility
    - people are used to it
  - Cons:
    - single point of failure at the commander
    - less or no sense of ownership by everyone
- Communion model
  - Pros
    - a community of leaders, each in his/her own domain
    - inherent sense of ownership
  - Cons
    - people aren't used to it (and this scares them)

27

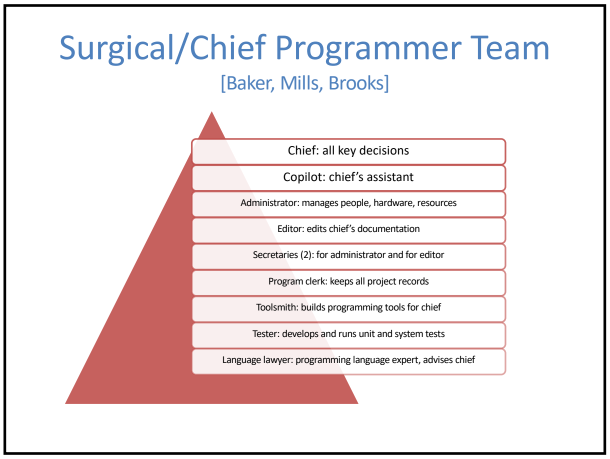


Team leadership

- Who makes the important product-wide decisions in your team?
  - One person?
  - All, by unanimous consent?
  - Other options?...

  - Is this an unspoken or an explicit agreement among team members?

28

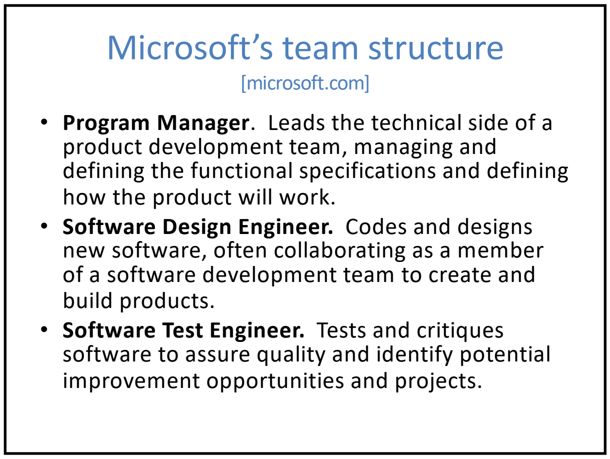


Surgical/Chief Programmer Team

[Baker, Mills, Brooks]

- Chief: all key decisions
- Copilot: chief’s assistant
- Administrator: manages people, hardware, resources
- Editor: edits chief’s documentation
- Secretaries (2): for administrator and for editor
- Program clerk: keeps all project records
- Toolsmith: builds programming tools for chief
- Tester: develops and runs unit and system tests
- Language lawyer: programming language expert, advises chief

29

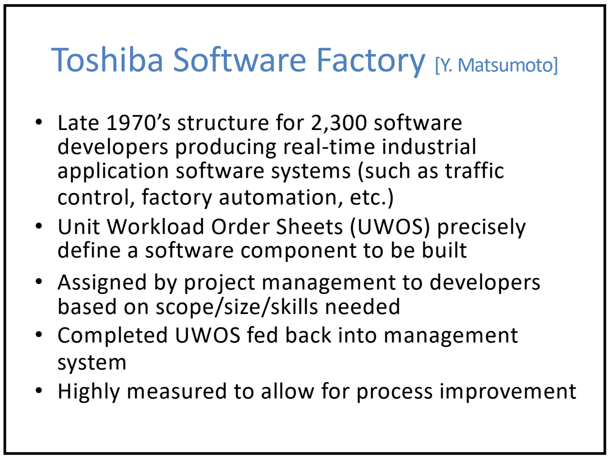


Microsoft’s team structure

[microsoft.com]

- **Program Manager**. Leads the technical side of a product development team, managing and defining the functional specifications and defining how the product will work.
- **Software Design Engineer.** Codes and designs new software, often collaborating as a member of a software development team to create and build products.
- **Software Test Engineer.** Tests and critiques software to assure quality and identify potential improvement opportunities and projects.

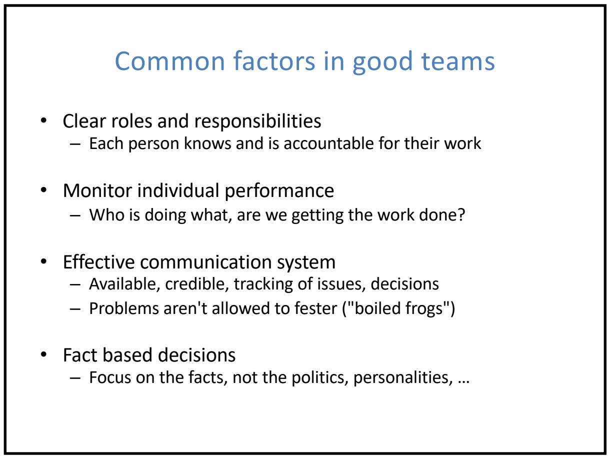
30



Toshiba Software Factory [Y. Matsumoto]

- Late 1970’s structure for 2,300 software developers producing real-time industrial application software systems (such as traffic control, factory automation, etc.)
- Unit Workload Order Sheets (UWOS) precisely define a software component to be built
- Assigned by project management to developers based on scope/size/skills needed
- Completed UWOS fed back into management system
- Highly measured to allow for process improvement

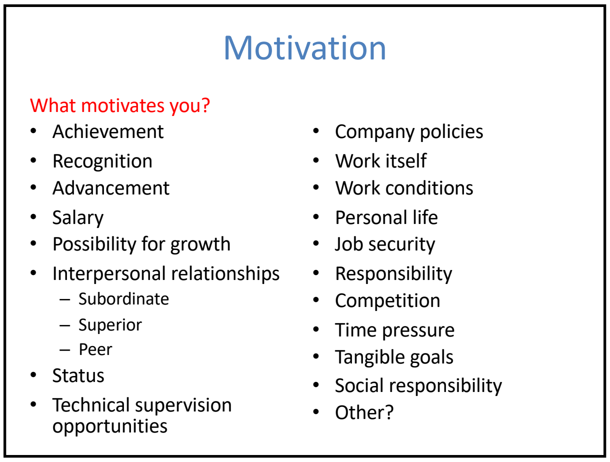
31



Common factors in good teams

- Clear roles and responsibilities
  - Each person knows and is accountable for their work
- Monitor individual performance
  - Who is doing what, are we getting the work done?
- Effective communication system
  - Available, credible, tracking of issues, decisions
  - Problems aren't allowed to fester ("boiled frogs")
- Fact based decisions
  - Focus on the facts, not the politics, personalities, …

32

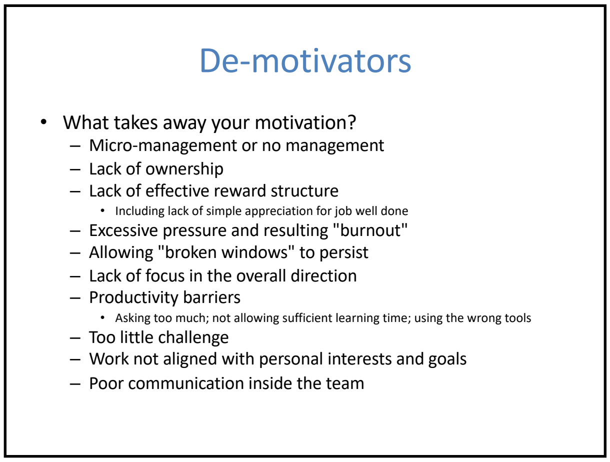


Motivation

What motivates you?

- Achievement
- Recognition
- Advancement
- Salary
- Possibility for growth
- Interpersonal relationships
  - Subordinate
  - Superior
  - Peer
- Status
- Technical supervision opportunities
- Company policies
- Work itself
- Work conditions
- Personal life
- Job security
- Responsibility
- Competition
- Time pressure
- Tangible goals
- Social responsibility
- Other?

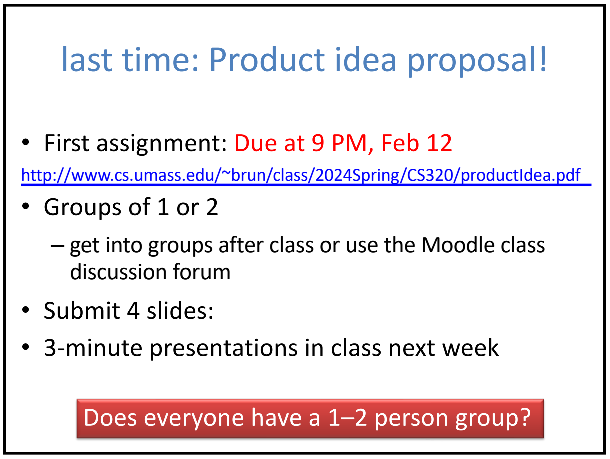
33



De-motivators

- What takes away your motivation?
  - Micro-management or no management
  - Lack of ownership
  - Lack of effective reward structure
    - Including lack of simple appreciation for job well done
  - Excessive pressure and resulting "burnout"
  - Allowing "broken windows" to persist
  - Lack of focus in the overall direction
  - Productivity barriers
    - Asking too much; not allowing sufficient learning time; using the wrong tools
  - Too little challenge
  - Work not aligned with personal interests and goals
  - Poor communication inside the team

34



last time: Product idea proposal!

- First assignment: Due at 9 PM, Feb 12

http://www.cs.umass.edu/~brun/class/2024Spring/CS320/productIdea.pdf

- Groups of 1 or 2
  - get into groups after class or use the Moodle class discussion forum
- Submit 4 slides:
- 3-minute presentations in class next week

Does everyone have a 1–2 person group?

35