“Standup” Progress Report

- **Standup Meeting**: efficiency technique promoted by XP/agile method
  - Goal of reducing time spent in meetings
  - Daily status meeting on agile project
    - What did I accomplish yesterday?
    - What will I do today?
    - What obstacles are impeding my progress?
- For projects: same questions on a weekly basis
  - Plan and hold at least one project meeting out of class
  - Choose a team name
  - Create team assemble page on wiki
  - Record meeting notes (Meeting Notes page)
  - Fill out Develop Logs

Review: Need to Organize the Work

- **Nature of a software project**
  - Software development produces a set of interlocking, interdependent work products
    - E.g. Requirements -> Design -> Code -> Test
  - Implies dependencies between tasks
  - Implies dependencies between people
- **Must organize the work such that**:  
  - Every task gets done
  - Tasks get done in the right order
  - Tasks are done by the right people
  - The product has the desired qualities
  - The end product is produced on time

Control and Risk

- **Risk**: a risk is defined as a condition that can lead to a loss of control
  - Incorrect, misunderstood, or missing requirements
  - Poor design choices
- Can lead to delivering wrong product, late, over cost..
- Well defined processes help organize work and control risks
Addressed by Software Processes

- Answers the “who”, “what”, “when”, etc. questions
  - What product should we work on next?
  - What kind of person should do the work?
  - What information is needed to do the work?
  - When is the work finished?
- **Software Process Model**: Abstract representation of a software life cycle as a set of
  1. Activities: tasks to be performed (how)
  2. Artifacts; work products produced (what)
  3. Roles: skills needed (who)

Projects Use Iterative Process

- Process viewed as a sequence of iterations
  - Essentially, a series of waterfalls
  - Each iteration builds on the previous one
  - Each iteration produces complete set of work products including deliverable software
- Addresses key risks (provides feedback, have something to deliver)

From Process to Plan

- Process definition manifests itself in the project plan
  - Process definition is an abstraction
  - Many possible ways of implementing the same process
- **Project plan makes process concrete**, it assigns
  - People to roles
  - Artifacts to deliverables and milestones
  - Activities to tasks over time
- Project plan should be one of the first products but expect it to evolve

Project Planning

- Goal: organize activities so the right people make decisions at the right time, and in a right order
- Part of the process
  - Activity: project planning
  - Artifact: the Project Plan
  - Roles: Project Manager (owner), team members
Project Plan

• Purpose: specifies how project resources will be organized to
  – Deliver on schedule
  – Address risks
• Audience: answers specific kinds of questions for specific types of users, e.g.:
  – Customers: When will the product be delivered?
  – Stakeholders: What is the development approach? How does it address project risks?
  – Managers: When will tasks be completed? What is the current progress against the plan?
  – Developers: What should I be working on and when?

Plan Outline

• Plan contents (template)
  – Purpose and audience: for the plan document
  – Project background
  – Team roles and responsibilities
  – Risks and risk mitigation
    • Should evolve with progress and understanding
  – Process: development process being used and its rationale
  – Mechanisms, methods, and techniques
    • Planning tools used, requirements methodology, design method
      IDEs, etc.
  – Detailed schedule and milestones*
  – Resources and references
• Usually owned by team manager
• Updated as project proceeds

Detailed Schedule and Milestones

• Maps people to tasks over time such that
  – On time delivery
  – People stay busy
• Answers: “Who is working on which tasks and when will they be finished?”
• Inputs
  – Set of artifacts to be created (superset of deliverables)
  – Precedence between tasks
  – People filling roles that perform tasks
  – Time budget for each task
• Output
  – Current project schedule
  – Deadline for each task
  – Sequencing among tasks
  – Allocation of people to tasks

Project Plan Template

• Use the template provided in your Assembla team workspace (under the Wiki tab)
• This should be a living document
  – Changed as the project progresses
  – Ideally, always gives a current view of the progress against the plan
    • Shows planned activities
    • Gives snapshot of the current project state
    • This is what I should see on your assembla site
**Project Planning Tools**

Work Breakdown Structure (WBS)
PERT Chart
Gantt Chart

---

**Work Breakdown Structure**

- This is a technique to analyze the content of work and cost by decomposing it into its component parts. It is produced by:
  - Identifying the key elements
  - Decomposing each element into component parts
  - Continuing to decompose until manageable work packages have been identified. These can then be allocated to the appropriate role/person
- The WBS is used to allocate responsibilities
- For the software, the WBS depends on the software architecture (discuss next)

---

**PERT Chart**

- Network analysis or PERT is used to analyze the relationships between the tasks identified by the work breakdown structure and to define the dependencies between tasks
- Helps identify where ordering of tasks may cause problems because of precedence or resource constraints
  - Where one person cannot do two tasks at the same time
  - Where adding a person can allow tasks to be done in parallel, shortening the project
PERT chart

If A -> B then A must finish before B starts.

Gantt Charts

- Method for visualizing a project schedule showing
  - The set of tasks
  - Start and completion times
  - Task dependencies
  - Responsibilities
- PERT charts can be reformatted as Gantt charts
- Resource: http://www.ganttproject.biz/
Example Gantt Chart

Project Milestone Planning

- Milestone planning is used to show the major steps that are needed to reach the goal on time.
- Milestones typically mark completion of key deliverables or establishment of baselines:
  - Baseline: when a work product is put under configuration management and all changes are controlled.
- Often associated with management review points:
  - E.g., Requirements baseline, project plan complete, code ready to test.
- Can use Gantt or PERT charts to show milestones.

A Simple Alternative

How much planning?

- How much planning is enough?
- Enough that:
  - Everyone knows what they should be doing.
  - Everyone knows what other people are supposed to be doing.
  - Everyone knows when specific tasks should be finished:
    - Specifically, they can track dependencies between their tasks and other peoples.
    - It is easy to determine the current status of the project against plan.
Walkthrough

- Consider: What kinds of questions should your documents answer?
  - Assume a manager unfamiliar with the project is reviewing your status
  - Would your documents answer key questions about the project goals and current status?
- Team page: Who is on the team?
- Project plan
  - Who is responsible for which tasks?
  - What are the anticipated risks and what are you doing about them?
  - What is your development process and how does it help address the risks?
  - What is the project schedule of tasks and deliverables?
  - What is the current status relative to schedule?
- ConOps: What capabilities will the software provide the user or customer?
- SRS: What are the detailed technical requirements?

Effective Meetings

Notes on effective meetings

- Only hold meetings if necessary
  - “Necessary” means that the most cost effective way to accomplish a goal is by meeting
- Have a goal, and a plan (agenda)
  - Clear meeting objectives
  - Known to all in advance (i.e., distribute via email)
- Plan to goal:
  - Participants - Invite only the necessary people
  - Schedule
  - Intended outcome
- Prepare
  - Cost of wasted time = Time x people x hourly cost
  - Cost of individual prep time is much less

Notes on effective meetings (2)

- Start on time, end on time
- Write down and disseminate the results
  - Leaves an audit trail of decisions
  - Makes people feel included
  - Limits the number of (informational) invitees
- End with concrete, specific action items
  - What must be done
  - Who should do it
  - What the follow-up is
Teamwork

What is a project team?

- Members of the project team make all the decisions going from problem to solution
- Intellectual control - making correct decisions about:
  - System requirements, system structure (decomposition), interfaces, detailed design, etc.
  - Realizes the functional and quality requirements
- Managerial control - making effective decisions about:
  - Overall budget and schedule, allocation of people to tasks, tasks to time, when tasks are complete
  - Realizes requirements for on-time and budget delivery

What is a Great Team?

- Diverse Skills
  - People skills, communication and writing skills, design skills, implementation skills and knowledge
- Coherence
  - Shared expectations
  - Ability to build and work toward a shared vision
- Mutual Respect and Responsibility
  - You don’t have to like each other, but you need to trust and respect each other — and to earn your teammates trust and respect
  - This is an enduring part of professionalism in the real world

Desired Skill Mix

- Ideal is a mix of skills: technical, communication, management
- At least one person with experience in team projects, preferably with some management experience*
- At least one person with strong skills in programming, program design, preferably including networking
- At least one person with strong communication skills and good written English
### Team Roles

- Manager: responsible for schedule
- Requirements analysts
- System architect
- Quality control (tester, reviewer)
- Technical writer
  - Technical documentation
  - User documentation (may be different skills)
- User interface designer
- Programmer
- Configuration control (build-master)

*Not 1-1 with people. Have a backup for each role.*

---

### What do software developers do?

- One way to measure: how do they spend their time?
- IBM study (McCue 78):
  - 50% team interactions
  - 30% working alone
  - 20% not directly productive

*Russian roulette*

---

### "Egoless" design

*(Weinberg, *Psychology of Computer Programming)*

- Investing ego in group
- "Letting go" of ego investment in code, design, ideas
  - No winning or losing design debates (focus on improving the product)
  - Once contributed, ideas belong to the group
  - Criticism is aimed at concepts, not people
- The best designers criticize their own designs!
  - Our own assumptions are the hardest to critique
  - Corollary: A good critic is your best ally
    - The hardest lesson to learn but one of the most valuable

---

### . . . but we are not egoless people

- Ego investment is normal
  - be aware of it, be in control of it
- Consider the egos of others
  - What are you attacking? Why?
  - What is motivation of the other person?
    - Are they feeling ignored? Not valued?
- Pride in accomplishments is OK, unless it interferes with accomplishment
Consensus decision making

- Consensus is not counting votes
  - Democracy is 51% agreement
  - Unanimity is 100% agreement
  - Consensus is neither
    - It is "buying in" by group as a whole, including those who disagree
- Everyone has their say
- Everyone accepts the decision, even if they don't prefer it
- Usually best approach for peer groups

*Consensus takes time and work, but is worthwhile*

Conflict

- Can be healthy and productive
- Can destroy a team if not carefully managed
- Manage conflict constructively
  - Soothe and protect egos
    - Everyone's job, but especially the manager's job
  - Keep conflict on a technical level (not personal)
  - Reward conflict resolution
- If team really cannot reach resolution, talk to instructor

Being a Good Team Member

- Attributes most valued by other team members
  - Dependability
    - When you say you'll do something, you do it
    - Correctly
    - On time
  - Carrying your own weight (doing a fair share of the work)
  - People will overlook almost everything else if you do these
- These are qualities we ask about in the peer evaluations at the course end

A Word on Managing

- A good manager supplies what is needed for the team to succeed. This includes (but is not limited to)
  - Resources
  - Planning and coordination
  - Pitching in when needed
  - Protection (especially from upper management)
  - Emotional support, etc.
- Good managers are are leaders not dictators (especially true for peer teams)
- Good managers are rare
Project Requirements: Iteration 1

- Iteration 1: basic functionality and capabilities
  - Aim for common capabilities (what everyone must implement)
  - Have all the major components that must communicate. Provide basic user services
- Think in terms of useful subsets
  - Build the smallest useful subset first: think about which capabilities will be needed by any future enhancements
  - Plan how you will add to it each increment

Questions?