### Student Startup Sequence
- Verify network connection
- Rotate to Landscape mode
- Start Presenter 2.0
- Maximize Application
- Role->Student
- Connect->Classroom 1

29 Jul 2005
CSE401, Summer'05

### Deliverables: Zero-Feature Release
- Build process, installation process, code repository, automated testing framework, bug tracking system
  - Maybe no tests yet and no tickets in the bug tracking system
- Installation package
  - Includes all of the items below
- Demo of one-step build and component communication
  - Checks out all sources from repository, compiles and builds binaries, packages them along with all existing documentation and automated tests, and places the result on a known site ready for downloading
  - Shows that your main components identified in the design can successfully communicate (be integrated) with each other
- Latest specification, design, and test plan documents
  - Keep them short! Consider what is / isn’t important for customers / devs.
- Up-to-date schedule
  - Includes what has been done and what remains to be done
- Release notes
  - Detailed instructions on how to run the demo
  - Known issues with prioritization

### Deliverables: Alpha/Beta Releases
- Installation package
  - Application sources and binaries
    - One-step build (for all sources) produces the installation package
  - Automated tests
    - Unit (single component) tests and acceptance (end-to-end) tests
  - Latest spec, design, and test plan documents
    - Keep it short! Consider what is / isn’t important for customers / devs.
- Release notes
  - Detailed instructions on how to run your application and/or a demo
  - Known issues with prioritization, expressed in a bug tracking system
- Up-to-date schedule
  - Include what has been done and what remains to be done.

Issues to consider:
- Who is your audience – customers or developers? What do they expect from this release? What defines success for them?

### Midterm Statistics

#### Midterm Results Histogram

- Other statistics:
  - Average: 31.64
  - Median: 30
  - Std Dev: 4.47
  - Max: 49
  - Min: 25

- Easiest Problems: 8, 12, 7
- Hardest Problems: 6, 9, 13

### Lecture 13: Risk Management

Valentin Razmov

### Outline
- The essence of risk and risk management
- Risk exposure and prioritization
- Coping with risks
- Risk assessment in practice – exercises
**Resources**
- *Rapid Development*, by Steve McConnell
- *Software Requirements*, by Karl Wiegers
- Lectures from winter 2005 and spring 2005

---

**Risks**

“If Las Vegas sounds too tame for you, software might be just the right gamble.”

— Steve McConnell

- *Risk* = a condition that could cause loss or otherwise threaten the success of a project

---

**Risk Management**

- The goal
  - Successful project completion
- The job
  - Identify the risks
  - Address the risks with specific actions
  - Avoid or resolve the risks before they become real threats to the project
- Remember this:
  - Mistakes are made on *every* project. The goal is to get to successful project completion *even though* mistakes were made.

---

**Levels of Risk Management**

- **Crisis management**
  - Address risks only after they have become problems
- **Fix on failure**
  - Address risks only after they have manifested
- **Risk mitigation**
  - Plan for when risks will show, but no attempt to prevent
- **Prevention**
  - Identify and prevent risks from becoming problems
- **Elimination of root causes**
  - Identify and eliminate factors that make risks possible

---

**Risks Can Be Related To...**

- Requirements
  - Misunderstanding requirements, inadequate user involvement, uncertain or changing project scope and objectives, continually changing requirements
- Design
- Testing
- Schedule
- Personnel
- Technology
- etc.

---

**The Multitude of Risks**

- McConnell gives a list of 111 (!) schedule risks.
  - This does *not* even include risks beyond scheduling.
  - How can one pay attention to all possible risks at once and proactively address them?
    - It’s a full-time job
      - Managers get paid very well when they are good at it.
    - Not all potential risks apply to all situations.
      - There are patterns; past experience or data on similar projects/teams can show what to pay extra attention to.
    - Not all risks that apply are equally important or likely.
      - Calls for risk prioritization
Risk Exposure

- Exposure = P(Loss) * |Loss|
  - E.g.: a 15% chance of slipping a project schedule by 10 weeks => a slippage time of 1.5 weeks is to be expected.
  - Allows a more intelligent estimate of the size of the "cushion" period you need for the project
  - Don't take the estimation too far!
    - It's not precise, after all.

Risk Prioritization

- Compute the risk exposure for each risk.
  - Sort all risks by their exposure: from high to low.
  - Move large-loss risks up on the list.
    - To avoid even unlikely catastrophic events
  - Address the risks starting from the top of the list.

Approaches to Coping with Risks

- Avoid the risk
- Transfer risk off the critical path
- Buy information
  - Bring in outside help
  - Prototype
- Publicize risk
- Schedule to accommodate some risk
- Monitor risks as project progresses

Risk Management in Practice: Team Construction

- Put yourself in the shoes of a hiring manager.
  - Task: Select 6 individuals from the list that follows to form a software team.
  - Product to build: A homework management and grading system.
  - Goal: Maximize the chance of project success.
  - Constraints: Budget limits your hiring choices.

Team Construction: Details of the Task

Three job classes to fill: PM, Developer, Tester
- Cost categories: A (expensive), B, C (cheap)
  - Constraints: employ at most 2 A's and at least 1 C
- Job class specific attributes
  - Communication – expected of PMs
  - Programming Ability – expected of Developers
  - Technology Understanding – expected of Testers
- Personality attributes
  - Charisma (leadership, external interactions)
  - Congeniality (getting along with team members)
  - Reliability
- Attribute ratings on a scale of 1 (low) to 5 (high)

Choose 6 team members: at most 2 A's, and at least 1 C

List the main factors affecting your choices: