Today ...

- Quiz 2
- Classic mistakes (cont.)
- Project charters
Four dimensions of development “speed”

People
- individual speed
- overall team dynamics

Product
- vaguely or clearly defined

Technology (language, tooling, etc.)
- may help or hinder development efforts

Process
- may utilize people’s time
- or create one stumbling block after another
Classic mistakes enumerated (McConnell, 96)

People ...

_Undermined motivation_

- many studies show motivation has a larger effect on productivity and quality than any other factor

In the case study:

- hokey pep talk at the beginning
- requiring overtime in the middle
- going on long vacation while team worked through holidays
- end-of-project bonuses working out to < $1 per overtime hour

_Weak personnel_

- individual capabilities (expertise) and team-member relationships can have 2nd greatest influence on productivity

In the case study:

- personnel selections made w.r.t. who could be hired fastest
- as opposed to who could get the most work done on the life of project
Uncontrolled problem employees

- failure to deal with problem personnel is most common complaint team members have of their managers

In the case study:

- team knew Chip was a bad apple, team lead didn’t do anything
- team ended up redoing all of Chip’s work

Adding people to a late project

- can be like pouring gasoline on a fire
- can take more productivity away from existing team members than it adds

Wishful thinking

- hoping something works with no basis for thinking it will

“None of the team members really believed that they could complete the project according to the schedule they were given, but they thought maybe if everyone worked hard, and nothing went wrong, and they got a few lucky breaks, they just might be able to pull it off.”

“We don’t need to show the final round of changes to the prototype to the customer. I’m sure we know what they want by now.”

“The team is saying that it will take extraordinary effort to meet the deadline, and they missed their first milestone by a few days, but I think they can bring this one in on time.”
Other people-related mistakes ...

- *Noising, crowded offices*
- *Friction between developers and customers*
- *Unrealistic expectations*
- *Lack of effective project sponsorship* (see charters)
- *Lack of stakeholder buy in* (see charters)
- *Lack of user input*
- *Politics over substance*

Process ...

*Abandon planning under pressure*

- abandon the plan, without a real alternative plan
- work then becomes uncoordinated and awkward

*Shortchanging “upstream” activities*

- aggressive schedules often cut “nonessentials”
- requirements gathering/analysis, architecture, design do not directly produce code ... so often seen as nonessential
- projects that skimp on these activities have to do the same work downstream at anywhere from **10 to 100 times** the cost
- better to do it right the first time!

The trick is how to do it “right”
Other process-related mistakes ...

- *Inadequate design* (rushed, alternatives not discussed, etc.)
- *Shortchanged quality assurance*
- *Insufficient management controls*
- *Premature or overly frequent convergence* (to final product)
- *Planning to catch up later*
- *Code-like-hell programming*
- *Insufficient planning and risk management*

Product ...

*Requirements “gold plating”* (complex, non-essential features)

**Feature creep**

*Developer gold plating* (stop when it is good enough)

*Push-me, pull-me negotiation*

*Research-oriented development*

- if product goals push state-of-the-art (algorithms, optimization, etc.)
- assume your scheduling is highly speculative

Technology ...

*Silver-bullet syndrome* (technology as “fix all”)

*Overestimated savings from new tools or methods*

- The team in the use case expected a 50% increase!

*Switching tools in the middle of a project*

*Lack of automated source-code control*
Moving forward

Now that we’ve talked about what can go wrong ...

We’ll focus on

• what we should do to develop software
• we’ll start with project charters and process models
Starting Projects

Q: What are problems that might happen just before a project starts?

Some problems with “unchartered” projects

- works starts without shared understanding of what is to be done
- semi-commitment of project resources (e.g., developers)
- project may have low “visibility” ... especially to upper management
- projects objectives and scope may change frequently (since unclear initially)
- unclear when the project is finished, or successful
- developers usually blamed if project is challenged or fails!

Project Charters

A written agreement about the project’s goals and scope

- written, but ideally short
- agreement between “stakeholders” (customers, management, developers)
- not a technical specification ...
Components of a project charter:

Project “vision” statement
- **why** are we doing in this project
- ... in terms of impact on users/customers

Project “mission” statement
- **what** will be done in this project
- ... a short, non-technical **summary** of the work

Project **business (external) objectives**
- desired outcomes of the project
- as concrete and measurable as possible
  - “Increase online sales by 15% next year”
  - “Improve customer response time by 10% in first 6 months”
- **not** specific tasks or deliverables of the project!

Project **internal objectives**
- goals of the team in terms of development process
- e.g., adopting new development tools or techniques
- also as concrete and measurable as possible
  - “Increase code reuse by 50% after first release”
- customer pays for this anyway, so should be open about it
The **committed resources**

- what is being invested to ensure project success
- money, people, training, products, etc.

The **risks**

- what are major risks to project success
- helps to ensure commitment, resources, etc.

Project “**boundary**” (or “**scope**”)

- what is inside the system/product and what is outside
- goal is to define the high-level system “boundary”
- often includes a **context diagram**

![context diagram](image)
Example mission statement (and some objectives) (Weigers, 2003)

“For scientists who need to request containers of chemicals, the Chemical Tracking System will provide a single point of access to the chemical stockroom and to vendors. The system will store the location of every chemical container within the company, the quantity of material remaining in it, and the complete history of each containers locations and usage. This system will save the company 25 percent on chemical costs in the first year of use by allowing the company to fully exploit chemicals that are already available within the company, dispose of fewer partially used or expired containers, and use a single standard chemical purchasing process. …”
The mission statement (and some objectives):

"For scientists who need to request containers of chemicals, the Chemical Tracking System will provide a single point of access to the chemical stockroom and to vendors. The system will store the location of every chemical container within the company, the quantity of material remaining in it, and the complete history of each container's locations and usage. This system will save the company 25 percent on chemical costs in the first year of use by allowing the company to fully exploit chemicals that are already available within the company, dispose of fewer partially used or expired containers, and use a single standard chemical purchasing process. …"