Principles of Software Construction: Objects, Design, and Concurrency
Part 7: Extra topics

SE: Toward People and Process

Christian Kästner    Charlie Garrod
Administrivia

• Homework 6 due tonight!
• Final exam Tuesday, May 5\textsuperscript{th}, 1 – 4 p.m. DH 2210
  – Final exam review session Sunday, May 3\textsuperscript{rd}, 4 – 6:30 p.m., Hamburg 1000
Key concepts from Tuesday
Pass, buy special tile GUI impl [ci skip]

fixed

master

Completed Exercise class

master

Checkpoint

master

Finish boom

master

GUI Finished, Test Cases Not Finished

master

New environment

master

Showing 1,862 changed files with 0 additions and 0 deletions.

Sorry, we could not display the entire diff because too many files (1,862) changed.
Today: First, a break to fill out FCEs

- [https://cmu.smartevals.com/](https://cmu.smartevals.com/)
Today: The next phase of SE: People and process
Compare: Software engineering to civil engineering
SE in 15-214

You?:
Compare: Software engineering to civil engineering
A GUIDE TO
UNDERSTANDING FLOW CHARTS
PRESENTED IN FLOW CHART FORM

START

DO YOU UNDERSTAND FLOW CHARTS?

YES

GOOD

NO

OKAY, YOU SEE THE LINE Labeled "YES"?

YES

...AND YOU CAN SEE THE ONES Labeled "NO"?

NO

LET'S GO DRINK.

6 DRINKS

HEY, I SHOULD TRY INSTALLING FREEBSD!

NO

Screw it.

6 DRINKS

WHY NOT?

WAIT, WHAT?

I HATE YOU.

NO

LISTEN.

BUT YOU JUST FOLLOWED THEM TWICE!

YES

(NO)

(BUT THAT WASN'T A QUESTION.)
Software project planning

Identify constraints
Estimate project parameters
Define milestones
Create schedule

Budget, Personal, Deadlines

Check progress
done?

yes

no

Reestimate project parameter
Refine schedule

Problem?

yes

Technical review

no

renegotiate constraints

new feature requests

Abort?

every 2-3 weeks
# Project planning

<table>
<thead>
<tr>
<th>Task</th>
<th>Duration (days)</th>
<th>Dependencies</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>T2</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>T3</td>
<td>15</td>
<td>T1</td>
</tr>
<tr>
<td>T4</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>T5</td>
<td>10</td>
<td>T2, T4</td>
</tr>
<tr>
<td>T6</td>
<td>5</td>
<td>T1, T2</td>
</tr>
<tr>
<td>T7</td>
<td>20</td>
<td>T1</td>
</tr>
<tr>
<td>T8</td>
<td>25</td>
<td>T4</td>
</tr>
<tr>
<td>T9</td>
<td>15</td>
<td>T3, T6</td>
</tr>
<tr>
<td>T10</td>
<td>15</td>
<td>T5, T7</td>
</tr>
<tr>
<td>T11</td>
<td>7</td>
<td>T9</td>
</tr>
<tr>
<td>T12</td>
<td>10</td>
<td>T11</td>
</tr>
</tbody>
</table>
Critical paths
Gantt diagrams
Resource planning
The "almost done" problem

% completed 90% 100%

time

planned actual

reported progress
Milestone trend analysis

Actual time

Estimated completion time
Risk management

1. Risk identification
   - List of potential risks

2. Risk analysis
   - Prioritized risk list

3. Risk planning
   - Risk avoidance and contingency plans

4. Risk monitoring
   - Risk assessment
Project planning

Identify constraints

Budget, Personal, Deadlines

Estimate project parameters

Define milestones

Create schedule

Check progress

Done?

no

yes

Reestimate project parameter

Refine schedule

Problem?

yes

no

renegotiate constraints

Technical review

Abort?

every 2-3 weeks

classic
Measurement in software engineering

• Performance, memory consumption, ...
• Productivity, complexity, ...
• Accuracy of predictions, ...
• Readability, maintainability, ...
• Stability, error rate, ...
• Usability, user acceptance, ...

• Depending on evaluation goal
  – Different data collection
  – Different metrics and scales
  – Different evaluations
Requirements engineering

- Knowledge acquisition: how to capture relevant detail about a system?
- Knowledge representation: once captured, how do we express it most effectively?
Beyond functional correctness

• E.g.:
  – Availability
  – Modifiability, portability
  – Performance, scalability
  – Security
  – Testability
  – Usability
  – Cost to build, cost to operate
Software engineering risks

• Project risks
  – Projects late, buggy, cost overruns
• System risks
  – Security and safety issues
  – e.g. The Toyota case
• Engineering risks
  – Unsuitable technology choices, validation issues, usability issues, scalability issues ...`
Teams

- Phases: Forming, Storming, Norming, Performing
- Problems: Groupthink, Social loafing, Communication, ...
Software crises...

Healthcare.gov: Government IT Project Failure at its Finest

Posted: 10/16/2013 6:33 pm

Week article on the Healthcare.gov failure is nothing if not instructive. From...
Next time...