Project Planning
and
Synchronizing Using Reviews

DSD Team

Outline

Implementing the process in a project plan
Importance of project planning in coordinating teams (duh)
Embedding the plan in the common workspace
Active Design Reviews
Using reviews for feedback and correction

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
Activities to tasks over time
Project plan should be one of the first products but expect it to evolve
For DSD, it is essential that distributed teams agree on the project plan

When will Foobar be ready?
Who is working on it?
Are there any problems?
Who should I tell about the problem I encountered in the Bazbot?
Project Plan

Minimal plan contents
- Risks and mitigation strategies
  - Evolves with progress and understanding
- Tasks to be performed
- Person(s) assigned to roles and tasks
- Deadline for each task
- Sequencing among tasks
  - Task dependencies
  - Development plan

Usually owned by team leads (one at each site)
Updated as project proceeds

On the Importance of Clearly Defined Roles

DSD coordination problems arise from communication problems
Lack of contextual information makes unclear
  - Exactly who knows what (who has expertise)
  - Exactly who is doing what (work allocation)
  - What questions or problems people have
  - What assumptions people are making
  - Etc.

Roles Help!

Well defined roles provide a badly needed structure
- Define who is responsible for what
- Gives guidance for expected expertise

Relations between roles tell you
- Who needs to talk to each other (e.g., shared responsibility, handoff, etc.)
- What you need to be talking about
- Provides bases for forming professional relationships

Upshot: in DSD it is critical that
1. Roles and their responsibilities are clearly defined
2. Well defined lines of communication are established between roles at different sites
3. People consistently perform their role's responsibilities

Template Link
Project Roles & Responsibilities

<table>
<thead>
<tr>
<th>Role</th>
<th>Number of team members taking on the role</th>
<th>Artifacts for which the role is responsible</th>
</tr>
</thead>
<tbody>
<tr>
<td>Systems Engineer</td>
<td>1</td>
<td>use cases, requirements, preliminary screenshots</td>
</tr>
<tr>
<td>Architect</td>
<td>1 or 2</td>
<td>Module, uses, process structures, interface specifications</td>
</tr>
<tr>
<td>Developer</td>
<td>&gt;1</td>
<td>Module implementation</td>
</tr>
<tr>
<td>Tester &amp; Integrator</td>
<td>&gt;1</td>
<td>Module tests, System generation and verification plan, test results report</td>
</tr>
<tr>
<td>Project Manager</td>
<td>1</td>
<td>Project plan, project measures, retrospective report</td>
</tr>
</tbody>
</table>

Work Breakdown Structure

Decompose work into component parts
Identify key elements
Decompose each element into component parts
Continue until manageable work packages
Allocated work packages to roles or people
The WBS is used to allocate responsibilities
For software, the WBS depends on the software architecture (discuss next)

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 charts.

A software project Gantt

Grouped as milestones
Work assignments
Parallel tasks (level of effort)
Precedence (constrained order)

You need a lot of detail (including work assignments) for the current increment; less for further increments.
DSD Project Plan

Common project plan is key to coordination
- Clear definition of roles and responsibilities
- Clear dependencies between tasks hence, what needs to be done next
- Provides basis for tracking progress

Just one part of necessary communication!
- Teams must agree on project plan but...
- Still easy to have misunderstanding about meaning of plan
- Still may go off track
  - Must detect and correct as soon as possible
  - This is not easy
- Plan must be continuously updated

Summary

Processes provide methods for managing software developments over time
- Must choose the right process to address a project’s specific problems and constraints
- Incremental process provide the feedback between distributed teams required for DSD
- Processes must be defined and realized in a project plan
- The project plan must evolve as the project progresses

Assignment

Objective: we should be able to look at your team web site at any time and clearly discern the following from the current plan:
- Who is supposed to be doing what
- Which deliverables are due next
- Current status of the work against the plan

This means you must collaborate to do the following
- Agree on roles
- Agree on regular tasks and deliverables
- Divide the work among sites
- Track the work at each site
- Post summary results at regular intervals

Active Reviews

Peer Reviews
Inefficiencies in Peer Review
Active Reviews
Peer Reviews

Peer Review: a process by which a software product is examined by peers of the product’s authors with the goal of finding defects

Why do we do peer reviews?
- Review is often the only available verification method before code exists
- Formal peer reviews (inspections) instill some discipline in the review process

Particularly important for distributed teams
- Supports communication and visibility
- Provides feedback on both quality and understanding
  - i.e., makes the communication effectiveness and level of understanding visible
- A good review shows communication is working!

Effectiveness of Peer Reviews

Generally considered most effective manual technique for detecting defects
- E.g., Bell-Northern found 1 hour code inspecting saves 2 to 4 hours code testing
- Effect is magnified in requirements inspections (e.g., 30 times in one study)

Means that you should be doing peer reviews, but...
- Doesn’t mean that manual inspections cannot be improved
- Doesn’t mean that manual inspections are the best way to check for every properties (e.g., completeness)
  - Should be one component of the overall V&V process

Peer Review Issues

Several inefficiencies, particularly for large or distributed projects
- Tendency for reviews to be incomplete and shallow
  - Reviewers typically swamped with information, much of it irrelevant to the review purpose
  - Reviewers lack clear individual responsibility
- Effectiveness depends on reviewers to initiate actions
  - Review process requires reviewers to speak out
  - Keeping quiet gives lowest personal risk
- Rewards of finding errors are unclear at best
- Process depends on group meetings
  - High overhead
  - Difficult to do for DSD
- No way to cross-check unstated assumptions
**Desired Qualities of Effective Review**

- Ensures adequate coverage of artifact in breadth and depth
- Reviewers review only issues on which they have expertise
- Review process is active: i.e., performing the review produces visible output (risk in in doing nothing)
- Individual responsibilities are clear and fulfilling them is evidence of a job well done
- Review process focuses on finding specific kinds of errors
- Small group communication in place of large meetings
  - Permit detailed discussion of issues
  - Expose where assumptions differ

**Active Review Process**

- **Role:** Quality Assurance
- **Activities:**
  - Identify several types of review each targeting a different type of error (e.g., UI behavior, consistency between safety assertions and functions)
  - Identify appropriate classes of reviewers for each type of review (specialists, potential users, methodology experts)
  - Assign reviews to achieve coverage: each applicable type of review is applied to each part of the specification
- **Design review questionnaires** (key difference)
- **Conduct review**
- **Revise**
- **Artifacts**
  - Input: documentation to be reviewed and support docs (e.g., module spec plus requirements/use cases)
  - Output: Review questionnaires

**Examples**

In practice: an active review asks a qualified reviewer to check a specific part of a work product for specific kinds of defects by answering specific questions, e.g.,

- Ask a designer to check the functional completeness by showing the calls sequences sufficient to implement a set of use cases
- Ask a systems analyst to check the ability to create required subsets by showing which modules would use which
- As a developer to check the data validity of a module’s specification by showing what the output would be for in-range and out-of-range values
- Ask a technical writer to check the SRS for grammatical errors

Can be applied to any kind of artifact from requirements to code as long as the reviewer is forced to us the document

**Conventional vs. Active Questions**

- **Goal:** Make the reviewer(s) think hard about what they are reviewing
  - Define questions that the review must answer by using the specification
  - Target questions to bring out key issues
  - Phrase questions to require “active” answers (not just “yes”)

<table>
<thead>
<tr>
<th>Conventional Design Review Questions</th>
<th>Active Design Review Questions*</th>
</tr>
</thead>
<tbody>
<tr>
<td>Are exceptions defined for every program?</td>
<td>For each access program in the module, what exceptions can occur?</td>
</tr>
<tr>
<td>Are the right exceptions defined for every program?</td>
<td>What is the the range or set of legal values?</td>
</tr>
<tr>
<td>Are the data types defined?</td>
<td>For each data type, what are *+ an expression for a literal value of that data type; * a declaration statement to declare a variable for that type; * the greatest and least values in the range of that data type?</td>
</tr>
<tr>
<td>Are the programs sufficient?</td>
<td>Write a short pseudo-code program that uses the design to accomplish (some defined task).</td>
</tr>
</tbody>
</table>
Individual Review Process

Role: [Specialist] Reviewer
- Specific expertise in problem domain, architecture design, etc.

Artifacts
- Input: artifact under review (e.g., SRS, module spec), prepared review questions
- Output: answers to questions, defects, issues

Activities
- Overview of artifact
- Review each artifact to answer questions
- Discussion of results
  - One-on-one or small group, for DSD can be on-line
  - Focus on discussion of issues identified in review
  - Purpose of discussion is understanding of the issue (not necessarily agreement)

Why Active Reviews Work

Focuses reviewer’s skills and energies where they have strengths and where those skills are needed
- Questionnaire allows reviewers to concentrate on one concern at a time
- No one wastes time on parts of the document where there is little possibility of return

Largest part of review process (filling out questionnaires) is conducted independently and in parallel
- Reviewers must participate actively but need not risk speaking out in large meetings
- Downside: more work for V&V (but can be productively pursued in parallel with document creation)

Role of Use Cases

Use cases or scenarios can be effectively used in active review
- Apply requirements scenarios to verify design against requirements
  - “Show the sequence of program calls that would implement use case C”
  - “Which modules would have to change to add feature F (a likely change)?”
- Conversely, can check properties ask the reviewer to construct scenarios
  - “What sequence of calls would result in an exception E?”

Simple Integer Stack

//Module Interface Spec
void push(int newitem); //push an integer on the stack
int pop() //Remove the top int from the stack
boolean isEmpty() //Returns true if the stack is empty

- What kinds of questions would one ask?
- Suppose we are handling order numbers LIFO and different subsystems handle different groups of order numbers in parallel?
Summary

Need to do reviews to find defects
Critical for distributed teams
  Provides another communication pathway
  Makes level of understanding visible
Active reviews are more efficient and effective but may take more effort
Assignment
  Choose one artifact (e.g., a module spec)
  Review within your own team
  Request a review from PKU side (I’ll ask them to do the same)

Review Questions For Abstract Interfaces:

Requirements Sufficiency

Does the set of services provided specify all of the services that will be needed by users of this module? Are there any services defined that are not identified in the requirements?

Does the set of services needed specify all of the services that this module will need from other modules in order to operate correctly? What services are needed that are not identified in the requirements?

Review Questions For Abstract Interfaces:

Requirements Validity

For each service provided by the module, is the service valid for all expected uses of this module? If not, give an example of a use where the service is not valid.

For each service provided by the module, is the service valid for all expected versions of this module? If not, give an example of a needed configuration or version where the service is not valid.

For each service needed described in this specification, is a module (or set of modules) identified that this module is allowed to use to satisfy the need?

Review Questions For Abstract Interfaces:

Consistency Between Services Provided and Access Programs

For each Services Provided described in this specification, which access program(s) can be used to satisfy the service?

For each access program and signal specified in sections 2 and 6 which Service Provided is satisfied by the access programs?
Review Questions For Abstract Interfaces:

*Access Program Adequacy*

Is the set of access programs and signals sufficient to satisfy the uses needs of modules that are allowed to use this module?

Are there access programs that should be combined into one access program?

Are there single access programs that should be refactored into several different access programs?

Are the performance requirements adequate for the uses that will be made of this module?

---

Questions?

---

**Extras**

---

**Plans vs. Reality**
*(Rational vs. Irrational)*

---
Plans vs. Reality
(Rational vs. Irrational)

Reality

Plan

Plans vs. Reality
Making The Reality Seem Rational

Document as if it had been rational
Readers can follow a sequential story
Explain significant changeable decisions
What alternatives were (are) there?
Why did we choose A’’ rather than A or A’
Later readers (maintainers) understand the trade-offs and can be guided by them

Plans vs. Reality
Documenting The Decisions