Chapter 5: *Planning*

- Planning is a key activity
  - But the scope of planning activities varies among people involved in a WebE project.
    - A team leader plans, monitors, and coordinates the combined work of a WebE team.
    - A Web engineer manages day-to-day work—planning, monitoring, and controlling technical tasks.
- Take an agile approach to planning
  - Adapt effort and time spent on planning to the complexity of the WebApp increment

Planning guidelines

- Understand scope *before* you define work tasks or schedule for an increment
- Refine framework actions and tasks
- Be sure you have the right team
- Evaluate risks
- Define a schedule
- Identify quality filters
- Identify how you'll manage change

Project Task Network Model

- A project consists of 8 activities named A to H.
- Construct a *network model* so as to satisfy the *scheduling* requirements shown in the table below.
- Activity Completion time (days) Immediate predecessor activities

Activity	Completion time (days)	Immediate predecessor activities
A	3	None
B	6	A
C	7	A
D	5	A
E	13	B,C
F	8	C,D
G	11	D,F
H	6	G,E

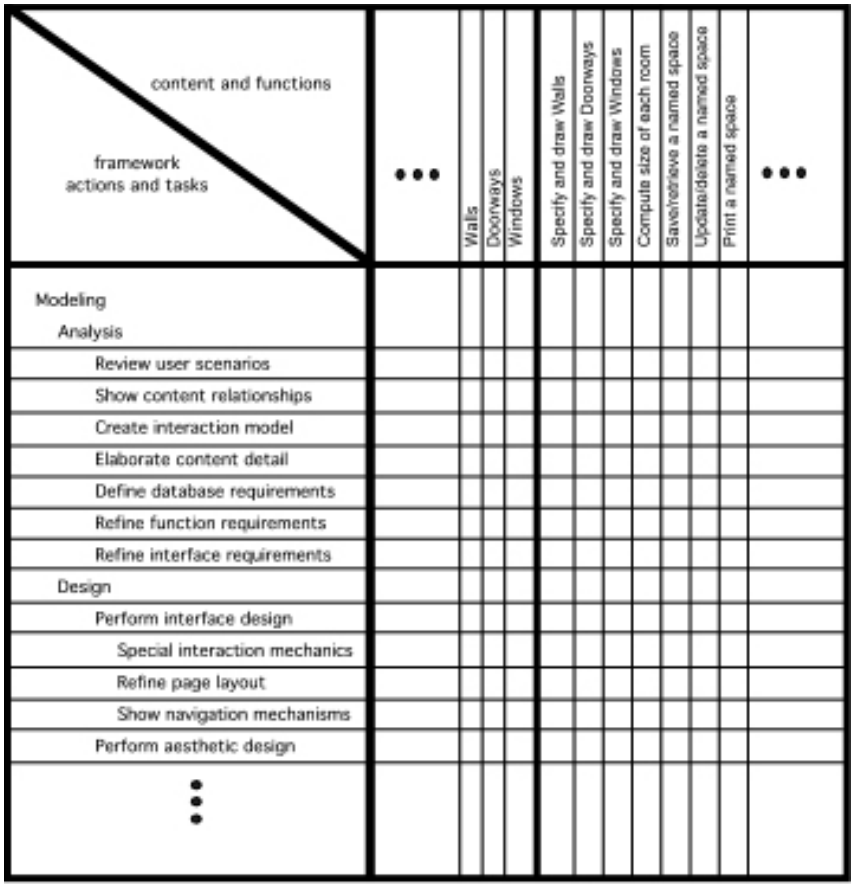
Create a network diagram of the system; use activity-on-node notation.

- Find *the least time required to complete* the whole project.
- List the project's *critical activities*.
- List the *float-time* associated with each of the project's non-critical activities.
- How is the project completion time affected if:
- activity F is delayed by 3 days
- activity E is delayed by 7 days
- activity G is finished 7 days early

WebApp Project Scope

- To plan effectively, you need to understand project scope
- To establish scope be sure you understand:
  - **Context.**
    - How does the WebApp fit into a business context, and what constraints are imposed as a result of the context?
  - **Information objectives.**
    - What customer-visible content objects are used by the WebApp increment?
  - **Functionality.**
    - What functions are initiated by the end user or invoked internally by the WebApp to meet the requirements defined in usage scenarios?
  - **Constraints and performance.**
    - What technical and environmental constraints are relevant?
    - What special performance issues (including security and privacy issues) will require design and construction effort?

Refining Actions and Tasks



The Team

- *Interestingly, people working together with **good communication** and **interaction** can operate at noticeably higher levels than when they use their individual talents. We see this time and again in brainstorming and joint problem-solving sessions. Therefore, **agile project teams [WebE teams] focus on increasing both individual competencies and collaboration levels**. [*Cockburn and Highsmith**]***
- But how do successful teams conduct their business?
  - A set of team guidelines should be established.
  - Strong leadership is a must.
  - Respect for individual talents is critical.
  - Every member of the team should commit.
  - It's easy to get started, but it's very hard to sustain momentum.

Managing Risk

- Many problems can arise during a project
- Risk management focuses on understanding and managing these problems
  - *Identify the risk; Assess its probability of occurrence; Estimate its impact; Establish a contingency plan*
- Considers risk at two different levels of granularity

  (1) the impact of risk on the *entire WebApp project*, and

  (2) the impact of risk on the *current WebApp increment*
- Typical risks:
  - Is the timeframe defined and reasonable?
  - Will the increments provide ongoing value for end users
  - How will requests for change impact delivery schedules?
  - Is the available technology appropriate for the job?
  - Does the team have the right mix of skills to build this increment

Identifying Risks

- Address the fundamental question: *"What can go wrong?"*
- Each team member is asked to make a list of risks
  - *People risks*: potential problems that can be directly traced to some human action or failing.
  - *Product risks*: potential problems associated with WebApp content, functions, constraints, or performance.
  - *Process risks*: problems that are tied to the framework actions and tasks that have been chosen by the team

Risk Analysis

Risks	probability	impact
People		
Little XML experience on team	80%	3
Stakeholders uncooperative	60%	2
Senior manager may change mid-stream	40%	1
Product		
Informational content may be outdated	50%	2
Algorithms may not be adequately defined	80%	3
Security for WebApp more difficult than expected	80%	3
Database integration more difficult than expected	40%	3
Space def. capability more difficult than expected	70%	3
Process		
Not enough emphasis on communication	60%	2
Too many analysis tasks (too much time spent)	30%	1
Not enough emphasis on navigation design	40%	2
⋮	⋮	⋮

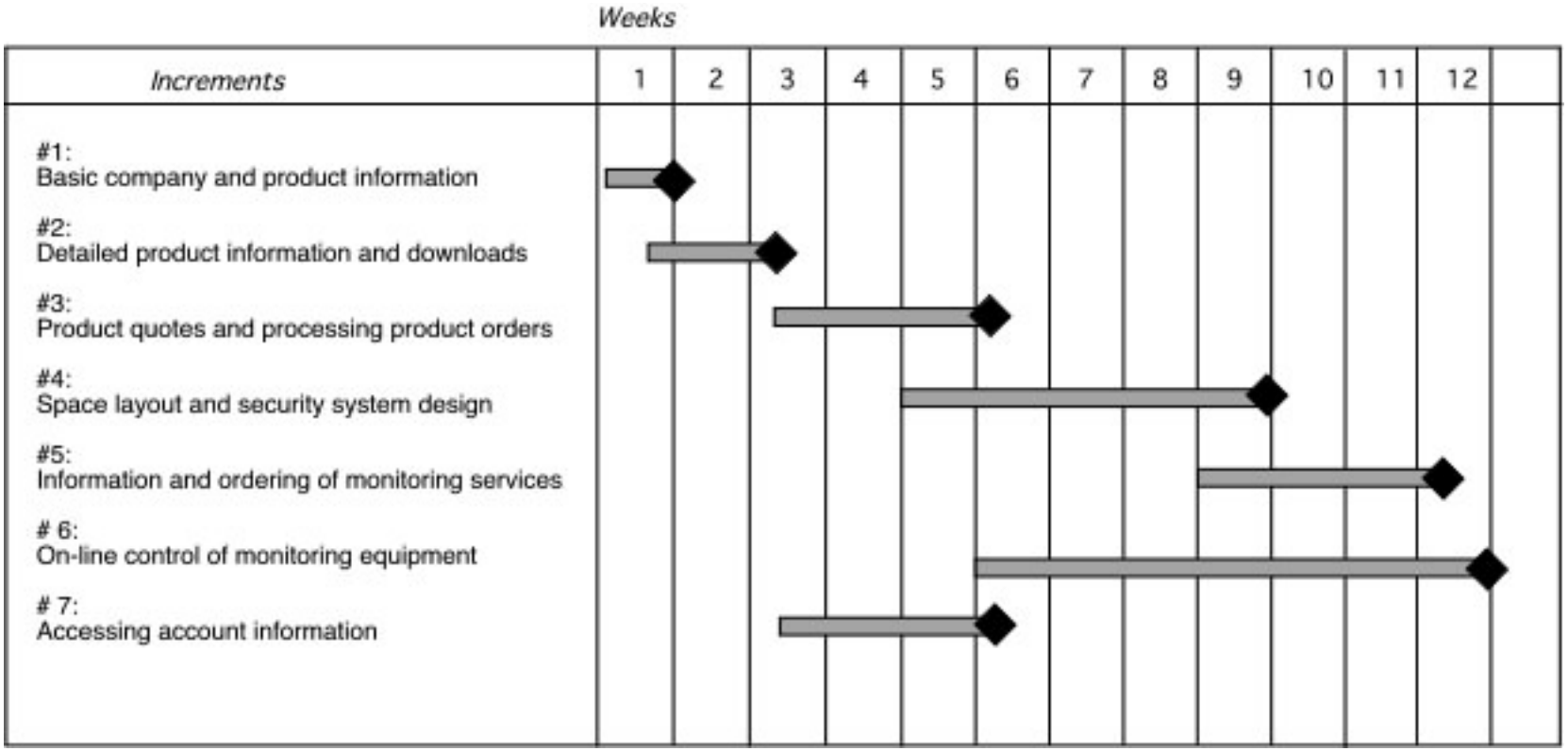
Risk Contingency Planning

- Development time spans are short, so *contingency plans are usually not formally documented*.
    - Document as information notes by the team leader
- Consider each risk that falls above the ***cutoff line*** in the risk table and answer three questions:
  1. How can we *avoid* the risk altogether?
  2. What factors can we *monitor* to determine whether the risk is becoming more or less likely?
  3. Should the risk become a reality, what are we *going to do* about it?

Developing a Schedule

- How do projects fall behind schedule?
  - *One day at a time*, Fred Brooks
- Approach:
  - List all WebE actions and *tasks* for an increment
  - Build a *network* that depicts interdependencies
  - Identify *tasks that are critical*
  - Track progress (especially critical tasks)
- The WebApp schedule evolves over time.
- During the first iteration a macroscopic schedule is developed.
  - Identify all increments and dates on which each will be deployed.
- For each subsequent increment
  - The entry for the increment on the macroscopic schedule is refined into a detailed schedule.

The Schedule



Estimating Time and Effort

- Two viable (and quick) approaches to detailed estimation

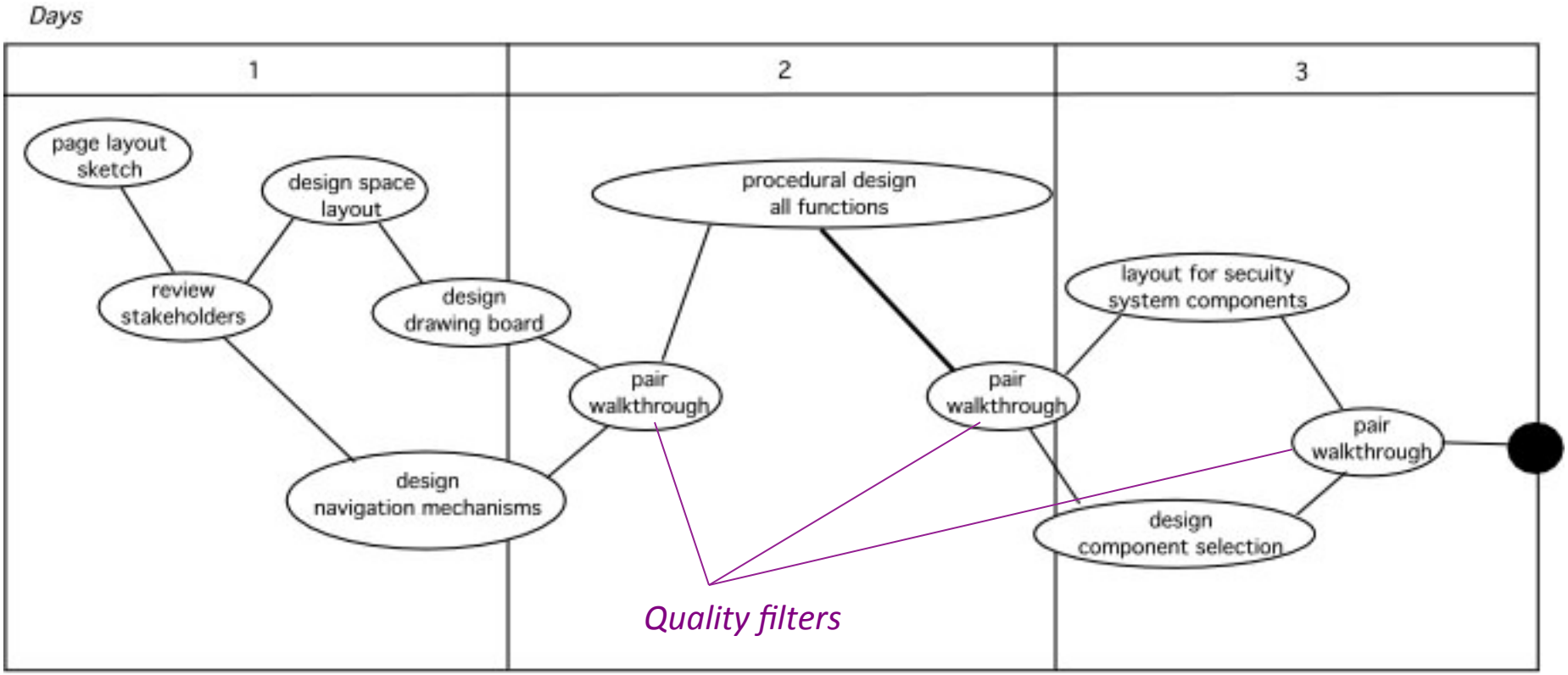
- *Usage scenario–based estimation*
  - Examines the usage scenarios for the increment, compares them to previous data on average effort (E_{avg}) for previous increments.
  - Adjust estimates based on perceived complexity
- *Product-process table*
  - *First column* lists, for all major WebE actions. Content objects and functions for an increment listed in *first row*.
  - Subsequent columns lists estimates of effort (in person-days) required to perform each of the main WebE actions for each content object and function.

- Warning! The relationship between effort, people and time is NOT linear.

Managing Quality

- What Quality Assurance Mechanisms Can the Team Use?
  - A thoughtful, thorough communication activity
  - Careful requirements gathering
  - Pair walkthroughs to assess the quality of all work products
  - Create a generic checklist that you can use to assess models
  - Use tests to evaluate quality

Quality Filters



Pair Walkthrough

- **Review the product, not the producer.**
- **Set an agenda and maintain it.** One of the key maladies of meetings of all types is *drift*. A walkthrough should be kept *on track and on schedule*.
- **Limit debate and rebuttal.** When an issue is raised by a reviewer, there may not be agreement on its impact. Rather than spending time debating the question, the issue should be recorded for resolution later.
- **Enunciate problem areas, but don't attempt to solve every problem noted.** A walkthrough is *not a problem-solving session*.
- **Take written notes**. Notes may be entered directly into a notebook computer.
- **Spend enough time to uncover quality problems, but not one minute more**. In general, a team walkthrough should be completed within *60 to 90 minutes at the most*.

Change Management

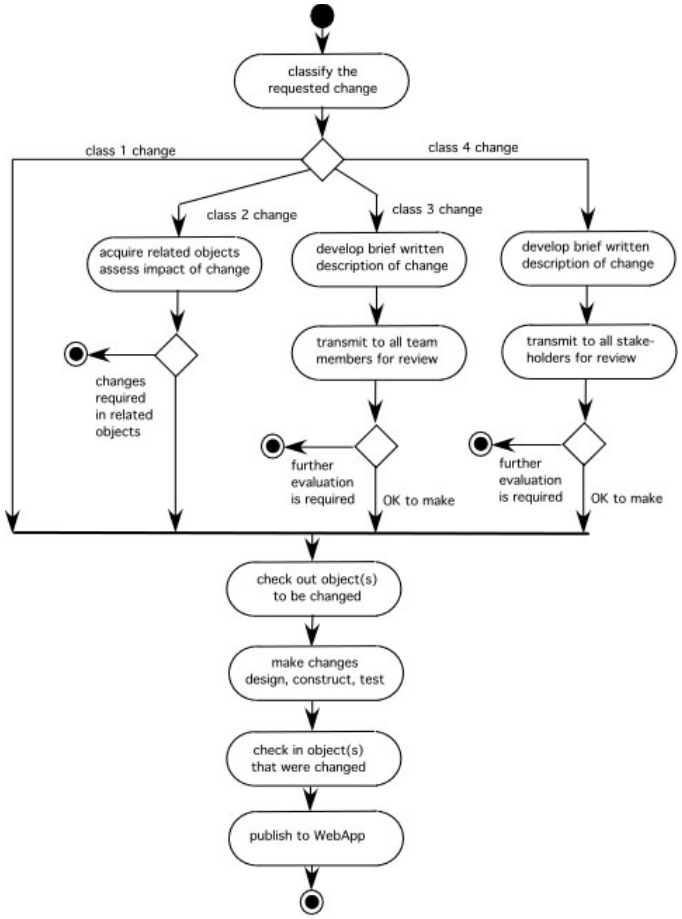


Figure 5-7 Change management for WebApps