Planning

Chapter 5

During Communication Activity We defined Scope

Scope is defined by answering the following questions:

- **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 and produced 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 will impact the framework activities that follow? What special performance issues will require design and construction effort?

communication **work products** relevant to the planning activity.

- Statement describing business motivation for the overall WebApp
- Statement of overall objective for the WebApp
- List of user categories
- List of informational goals for the WebApp increment to be planned
- List of applicative (functional) goals for the WebApp increment to be planned
- Description of the increment (the statement of scope)
- List of content objects for the increment
- List of functions for the increment
- Set of usage scenarios that describe how each user category will interact with the increment

What If Gaps Still Exist in Your Understanding?

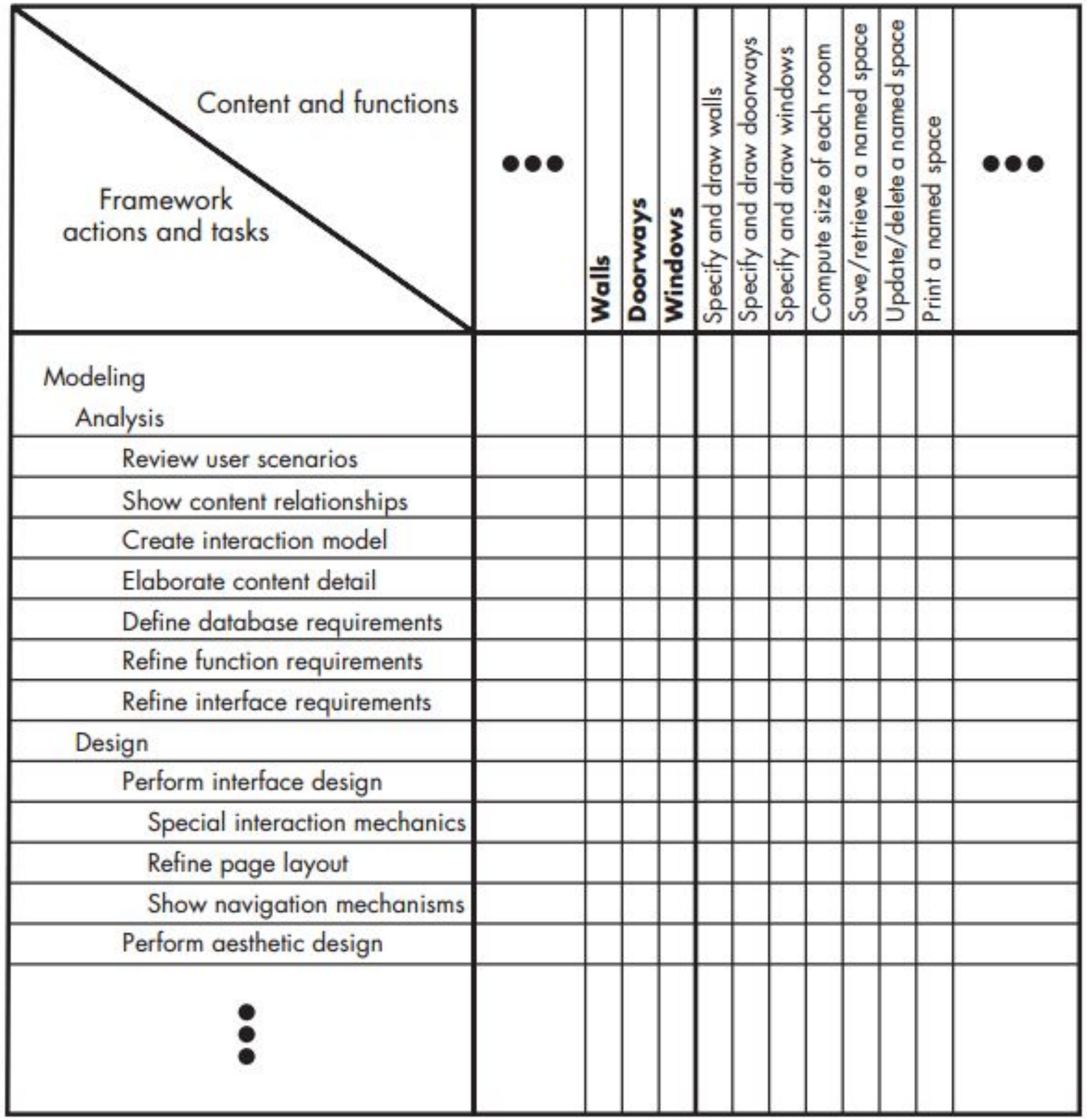
- You’ll have to accept the fact that things remain a bit uncertain,—it’s one of the risks inherent in all engineering work.
- You’ll have to complete the planning activity with imperfect information and move on.

What Actions and Tasks Are Required?

- create a task table for

Scenario

- Develop a layout of the space to be monitored. [page 72]



Content and functions / Framework actions and tasks	•••	**Walls**	**Doorways**	**Windows**	Specify and draw walls	Specify and draw doorways	Specify and draw windows	Compute size of each room	Save/retrieve a named space	Update/delete a named space	Print a named space	•••
Modeling Analysis												
Review user scenarios												
Show content relationships												
Create interaction model												
Elaborate content detail												
Define database requirements												
Refine function requirements												
Refine interface requirements												
Design												
Perform interface design												
Special interaction mechanics												
Refine page layout												
Show navigation mechanisms												
Perform aesthetic design												
⋮												

What Work Products Will Be Produced?

- intermediate work products
  - modeling representations,
  - interface sketches,
  - navigation maps

should be kept to the minimum that is necessary to provide appropriate guidance for the next framework action or task

What Is the Appropriate Way to Assess Quality?

- If a WebE team stresses quality in all framework activities, the team reduces the amount of rework that it must do

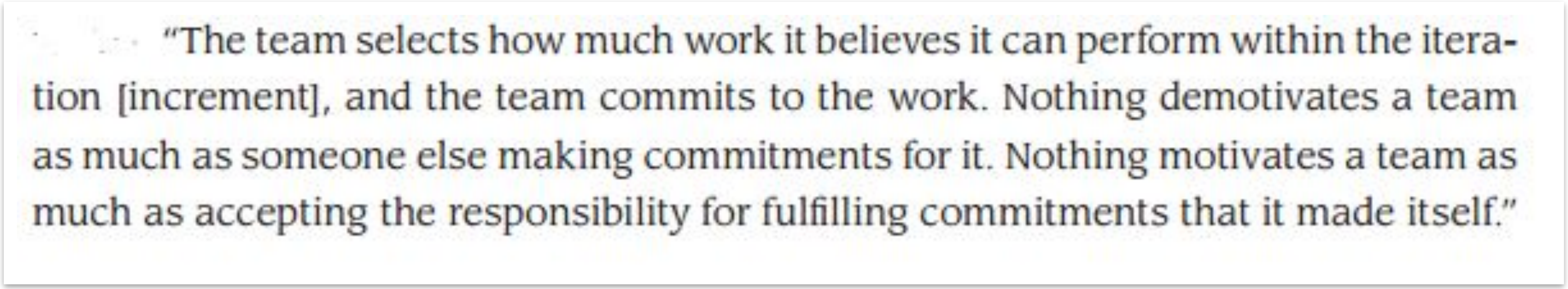
- You must explicitly define meaning of “WebApp quality” and define a set of tasks that will help ensure high quality

- WebApp quality
  - completeness and accuracy of the problem definition,
  - the commodity of the solution design,
  - the firmness of construction, and
  - the overall degree of satisfaction to which the WebApp increment meets the needs of all stakeholders

Why Don’t Teams Jell and What Can Be Done to Help?

- **frenzied work atmosphere** in which team members waste energy and lose focus on the objectives of the work to be performed.
- **High frustration** caused by personal, business, or technological factors that causes friction among team members.
- “**Fragmented or poorly coordinated procedures**” or a poorly defined or improperly chosen process model that becomes a roadblock to accomplishment.
- **Unclear definition of roles** resulting in a lack of accountability and resultant finger-pointing.
- “**Continuous and repeated exposure to failure**” that leads to a loss of confidence and a lowering of morale.

Building a webE team



"The team selects how much work it believes it can perform within the iteration [increment], and the team commits to the work. Nothing demotivates a team as much as someone else making commitments for it. Nothing motivates a team as much as accepting the responsibility for fulfilling commitments that it made itself."

WebE teams should be *self-organizing*

- A self-organizing team has access to all information required to do the job, thereby avoiding a frenzied work environment in

- A self-organizing team has control over
  - the process that is employed,
  - the work products that are produced,
  - the work schedule that is defined, and
  - the quality and change management activities that are implemented.

  Therefore, the team avoids the frustration that occurs when there is a lack of control.

WebE teams should be *self-organizing*

- A self organizing team establishes its **own mechanisms for accountability** and defines a series of corrective approaches when a member of the team fails to perform.
- Every WebE team experiences small failures. The key to avoiding an atmosphere of failure is to **establish team-based techniques for feedback and problem solving**.
- failure by any member of the team must be viewed as a failure by the team itself. This leads to a **team-oriented approach to corrective action**, rather than the finger-pointing and mistrust that grows rapidly on toxic team

How can a team manage itself

- A team leader should be appointed to coordinate communication and work tasks
- assess progress and problems by conducting daily team meetings (15 to 20 minutes) to coordinate and synchronize the work that must be accomplished for that day.
- These brief meetings address four key questions:
    - What have we accomplished since the last meeting?
    - What needs to be accomplished before the next meeting?
    - How will each team member contribute to accomplishing what needs to be done?
    - What roadblocks exist that have to be overcome?

How Do We Build a Successful Team?

- 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.

Characteristics of a Good Team Leader

- **Motivation.** The ability to encourage technical people to produce to their best ability. This can be accomplished by providing incentives for high performance and imposing consequences for poor performance.

- **Organization.** The ability to mold existing processes (or invent new ones) that will enable the initial concept to be translated into a final product.

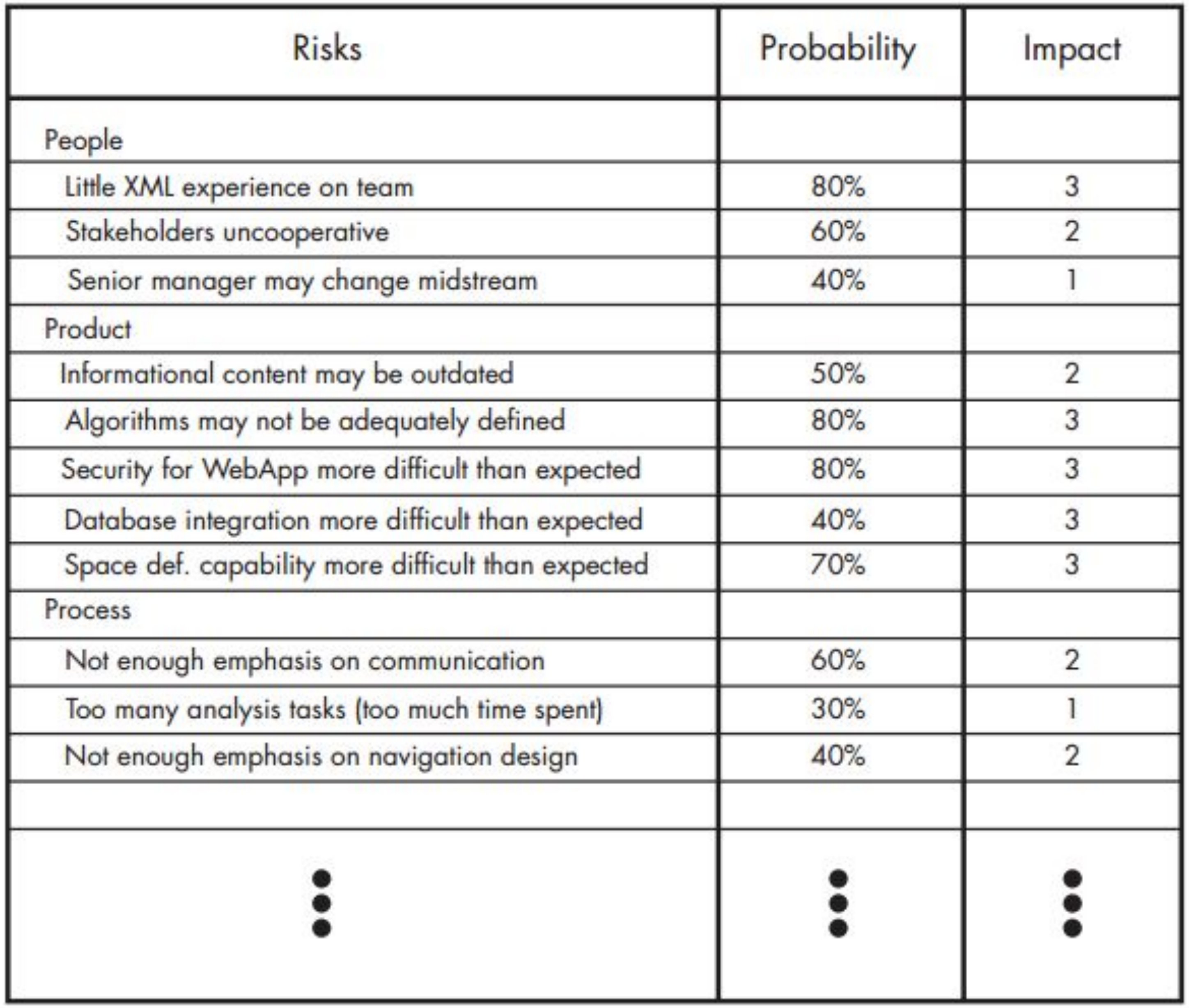
- **Ideas or innovation.** The ability to encourage people to create and feel creative even when they must work within bounds established for a particular WebApp

Managing Risk

- people risks
  - directly traced to some human action or failing
- product risks
  - can normally be traced to 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 Evaluation

- the likelihood or probability that the risk will become a reality,
- the consequences of the problems associated with the risk



Risks	Probability	Impact
People		
Little XML experience on team	80%	3
Stakeholders uncooperative	60%	2
Senior manager may change midstream	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
⋮	⋮	⋮

Contingency Plans

- How can we avoid the risk altogether?
- What factors can we monitor to determine whether the risk is becoming more or less likely?
- Should the risk become a reality, what are we going to do about it?

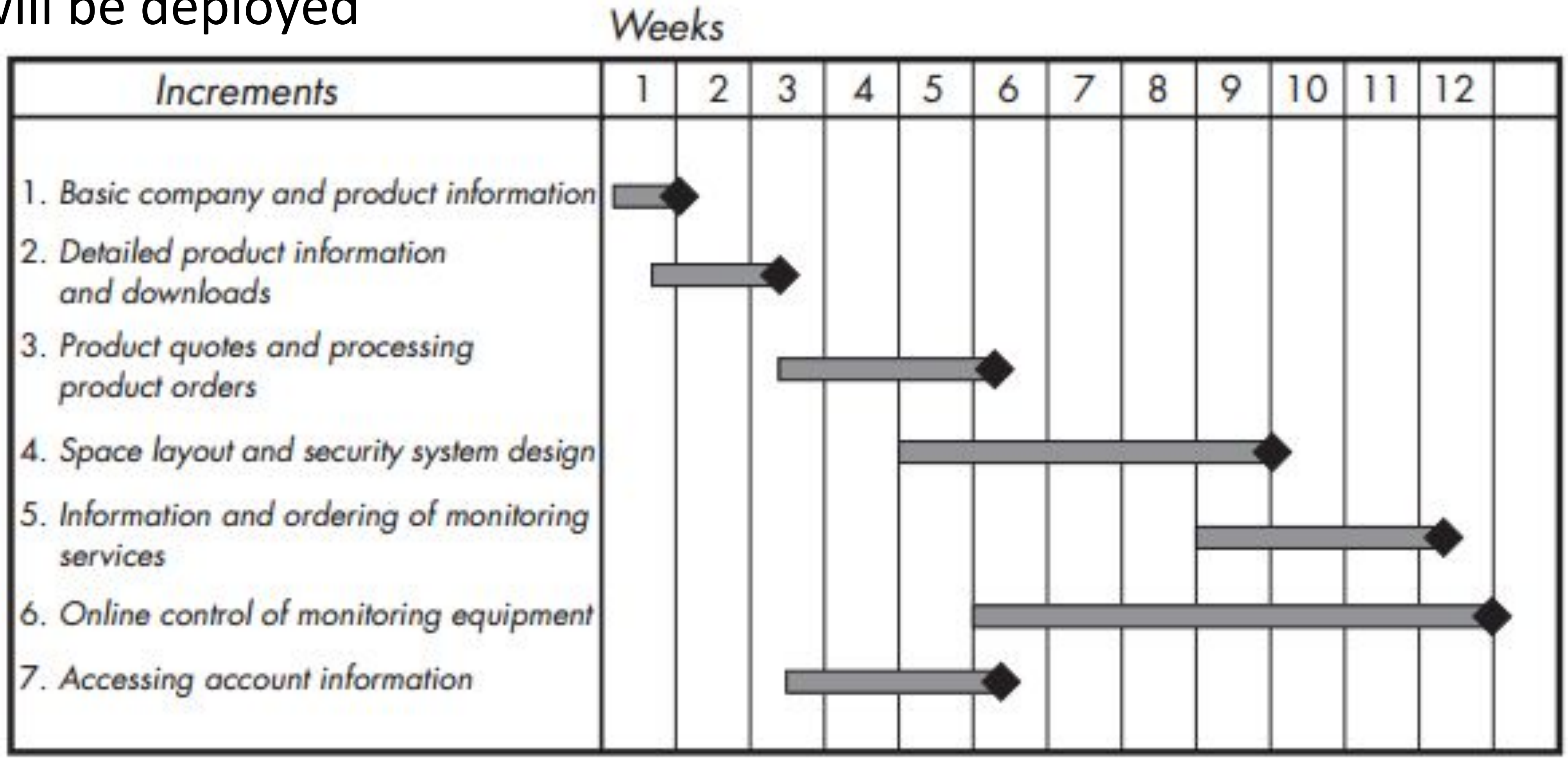
Developing Schedule

- *WebApp project scheduling* is an activity that allocates the estimated effort for specific WebE tasks across the planned time line (duration) for building an increment

- A Macroscopic schedule
- Incremental schedule.

Macroscopic Schedule

- identifies all WebApp increments and projects the dates on which each will be deployed



Incremental Scheduling

- Expand each task into fine grained tasks.

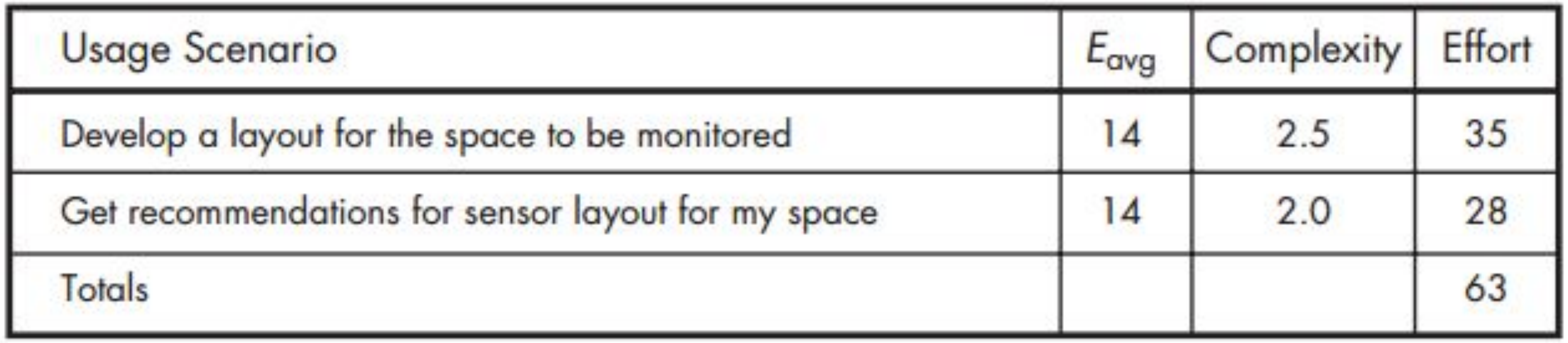
- *Design the interface*
  - Develop a sketch of the page layout for the space design page.
  - Review the layout with stakeholders.
  - Design the space layout navigation mechanisms.
  - Design the “drawing board” layout
  - Develop procedural details for the graphical wall layout function.
  - Develop procedural details for the wall length computation and display function.
  - Develop procedural details for the graphical window layout function.
  - Develop procedural details for the graphical door layout function.
  - Design mechanisms for selecting security system components (sensors, cameras, microphones, etc.).
  - Develop procedural details for the graphical layout of security system components.
  - Conduct pair walkthroughs as required

Estimating Effort and time

- *usage scenario–based estimation*
  - Examining the team's past history,
  - establish a value E_{avg}, the average effort (in person-days) required to deploy a usage scenario
  - Roughly Calculate the complexity of the scenario

Estimating Effort and time

Usage scenario–based estimation.

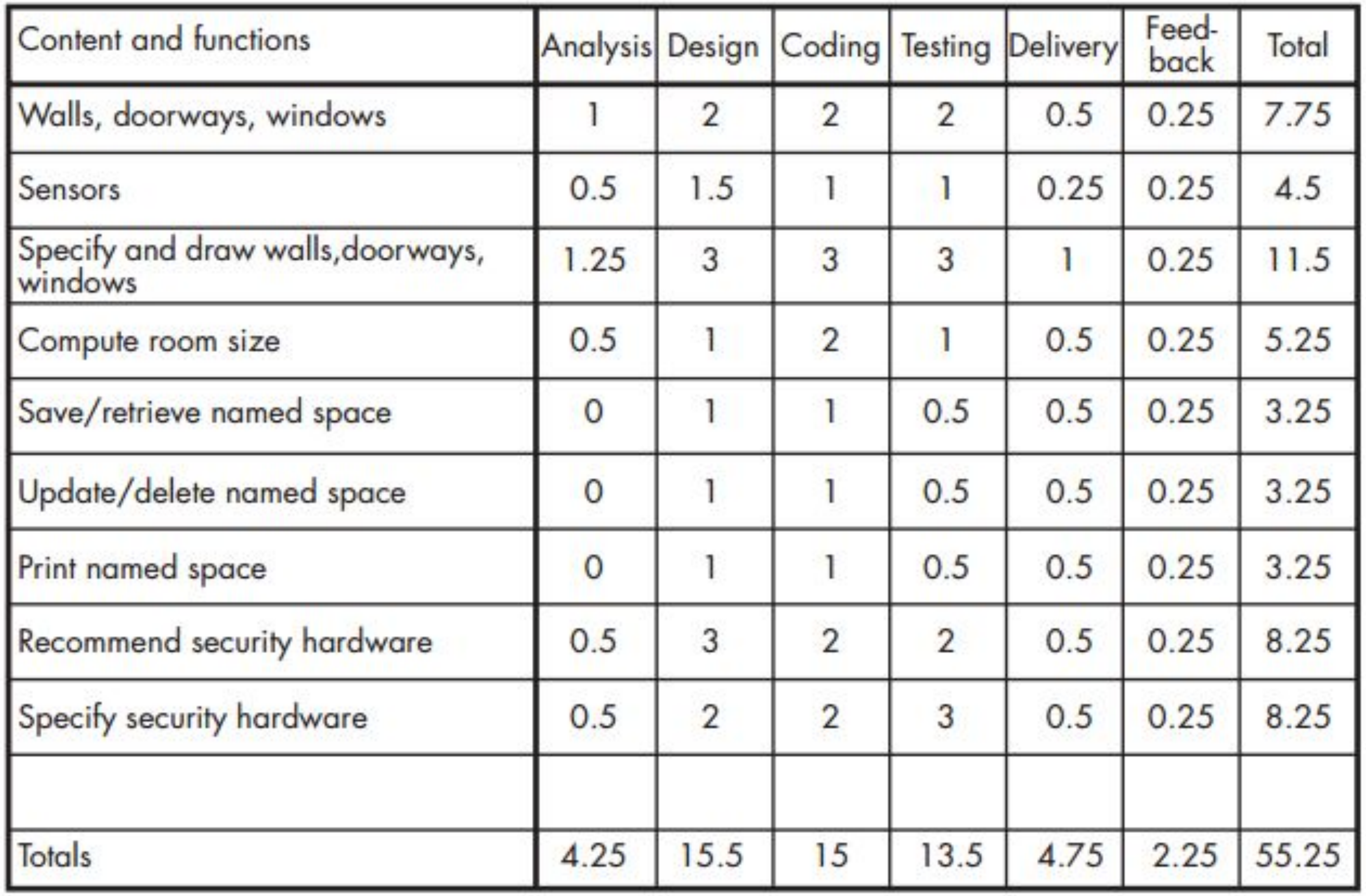


Usage Scenario	E_{avg}	Complexity	Effort
Develop a layout for the space to be monitored	14	2.5	35
Get recommendations for sensor layout for my space	14	2.0	28
Totals			63

Estimating Effort and time

- ***Product-process table***
  - all major WebE actions are listed in the first column of the table.
  - All major content objects and functions for an increment are listed in the first row.
  - Team members estimate the amount of effort (in person-days) required to perform the WebE action for each content object and function.

Estimating Effort and time



Content and functions	Analysis	Design	Coding	Testing	Delivery	Feed-back	Total
Walls, doorways, windows	1	2	2	2	0.5	0.25	7.75
Sensors	0.5	1.5	1	1	0.25	0.25	4.5
Specify and draw walls,doorways, windows	1.25	3	3	3	1	0.25	11.5
Compute room size	0.5	1	2	1	0.5	0.25	5.25
Save/retrieve named space	0	1	1	0.5	0.5	0.25	3.25
Update/delete named space	0	1	1	0.5	0.5	0.25	3.25
Print named space	0	1	1	0.5	0.5	0.25	3.25
Recommend security hardware	0.5	3	2	2	0.5	0.25	8.25
Specify security hardware	0.5	2	2	3	0.5	0.25	8.25
Totals	4.25	15.5	15	13.5	4.75	2.25	55.25

Managing Quality