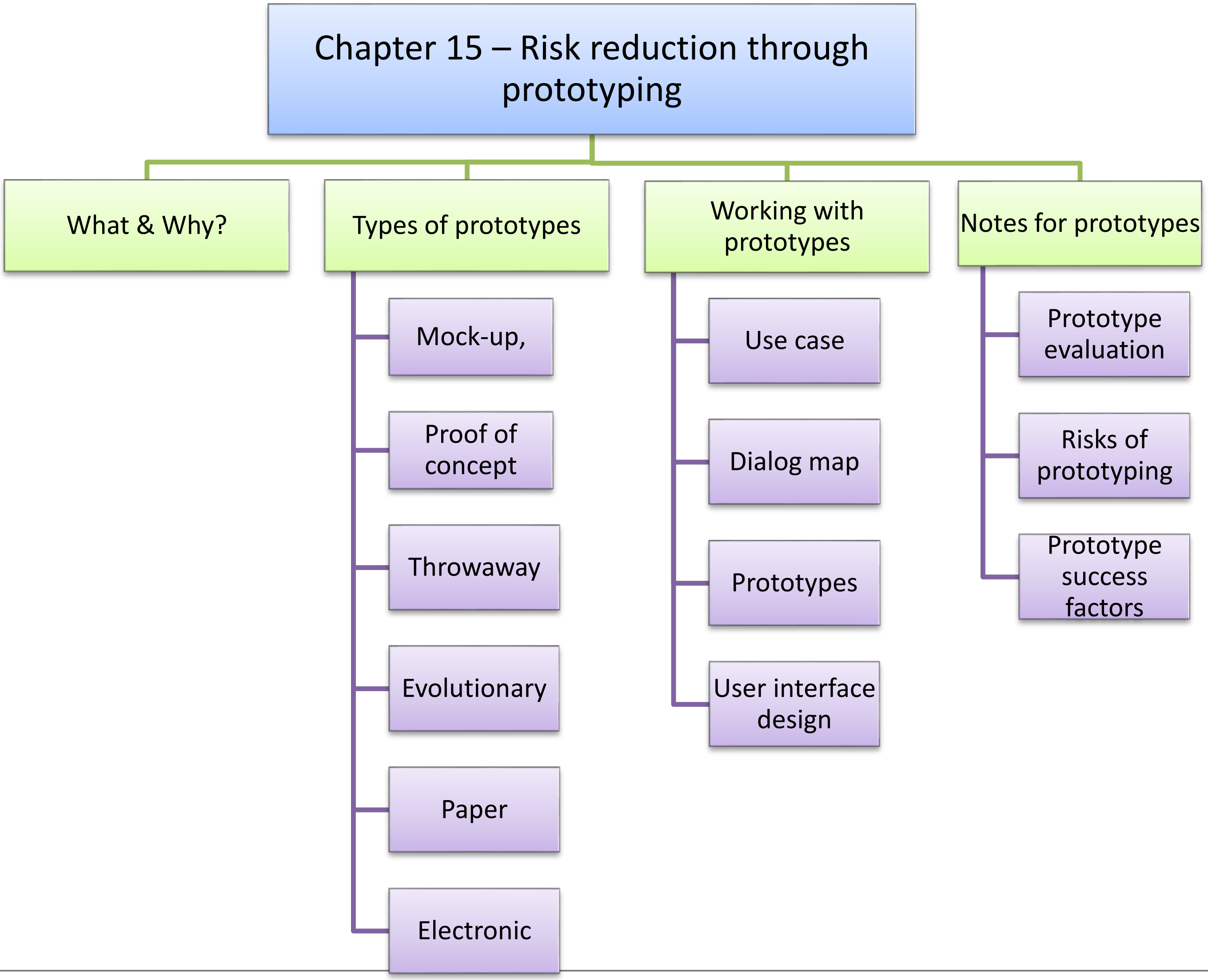
ĐẠI HỌC FPT CẦN THƠ

CHAPTER 16

First things first: Setting requirement priorities

cantho.fpt.edu.vn

Review chapter 15



Objectives

- This chapter discusses the importance of prioritizing requirements.
- Describes several prioritization techniques, and presents a spreadsheet tool for prioritization analysis based on value, cost, and risk.
- After finish this chapter, student should priority list of requirements basing on their projects.

Contents

1. Why prioritize requirements?
2. Some prioritization pragmatics (thực dụng).
3. Games people play with priorities.
4. Some prioritization techniques.
5. Prioritization based on value, cost, and risk.

Why prioritize requirements?

- Prioritization, also called *requirements triage* (chọn để làm) (Davis 2005), helps reveal competing goals, resolve conflicts, plan for staged or incremental deliveries, control scope creep, and make the necessary trade-off decisions.
- When customer expectations are high and timelines are short, you need to make sure the product delivers the most critical or valuable functionality as early as possible. Prioritization is a way to deal with competing demands for limited resources.
- Prioritization is a critical strategy for agile or other projects that develop products through a series of fixed-schedule timeboxes.
- Establish priorities early in the project, when you have more flexibility for achieving a successful project outcome, and revisit them periodically.

Some prioritization pragmatics

- Various stakeholders need to participate in prioritization, representing customers, project sponsors, project management, development, and perhaps other perspectives.
- You really need one ultimate decision maker when stakeholders can't agree.
- A good starting point is for the prioritization participants to agree upon a set of criteria to use for judging whether one requirement has higher priority than another.
- Alan Davis (2005) indicates that successful prioritization requires an understanding of six issues:
  - The needs of the customers
  - The relative importance of requirements to the customers
  - The timing at which capabilities need to be delivered
  - Requirements that serve as predecessors for other requirements and other relationships among requirements
  - Which requirements must be implemented as a group
  - The cost to satisfy each requirement

Games people play with priorities

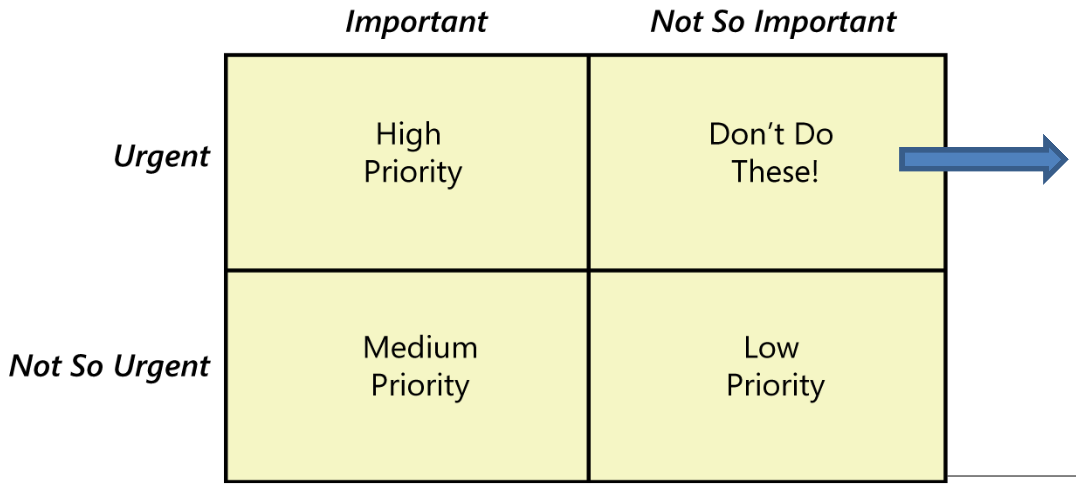
- The knee-jerk (ko suy nghĩ) response to a request for customers to set priorities sometimes is, “I need all these features. Just make it happen.” They feel that every requirement should be ranked as high priority, and they might not recognize that prioritization will help to ensure the project’s success.
- To encourage customers to acknowledge that some requirements have lower priority, the analyst can ask questions such as the following:
  - Is there some other way to satisfy the need that this requirement addresses?
  - What would the consequences be of omitting or deferring this requirement?
  - What effect would it have on the project’s business objectives if this requirement weren’t implemented for several months?
  - Why might a customer be unhappy if this requirement were deferred to a later release?
  - Is having this feature worth delaying release of all of the other features with this same priority?

Some prioritization techniques

- Several analytical and mathematical techniques have been proposed to assist with requirements prioritization.
- Some techniques are:
  - In or out:
    - The simplest of all prioritization methods is to have a group of stakeholders work down a list of requirements and make a binary decision: is it in, or is it out?
    - Keep referring to the project's business objectives to make this judgment, paring (cắt giảm) the list down to the bare minimum needed for the first release.
    - Then, when implementation of that release is under way (được thực hiện), you can go back to the previously "out" requirements and go through the process again for the next release.
  - Pairwise comparison (so sánh cặp) and rank ordering:
    - Rank ordering a list of requirements involves making pairwise comparisons between all of them so you can judge which member of each pair has higher priority.
    - Performing such comparisons becomes unwieldy (khó sử dụng) for more than a couple of dozen requirements.

Some prioritization techniques

- Some techniques are:
  - Three-level scale: A common prioritization approach groups requirements into three categories: High – Medium – Low
    - *High-priority* requirements are both important (customers need the capability) and urgent (customers need it in the next release).
    - *Medium-priority* requirements are important (customers need the capability) but not urgent (they can wait for a later release).
    - *Low-priority* requirements are neither important (customers can live without the capability if necessary) nor urgent (customers can wait, perhaps forever).



Requirements in the fourth quadrant appear to be urgent to some stakeholder, perhaps for political reasons, but they really aren't important to achieving the business objectives. Don't waste your time working on these, because they don't add sufficient value to the product.

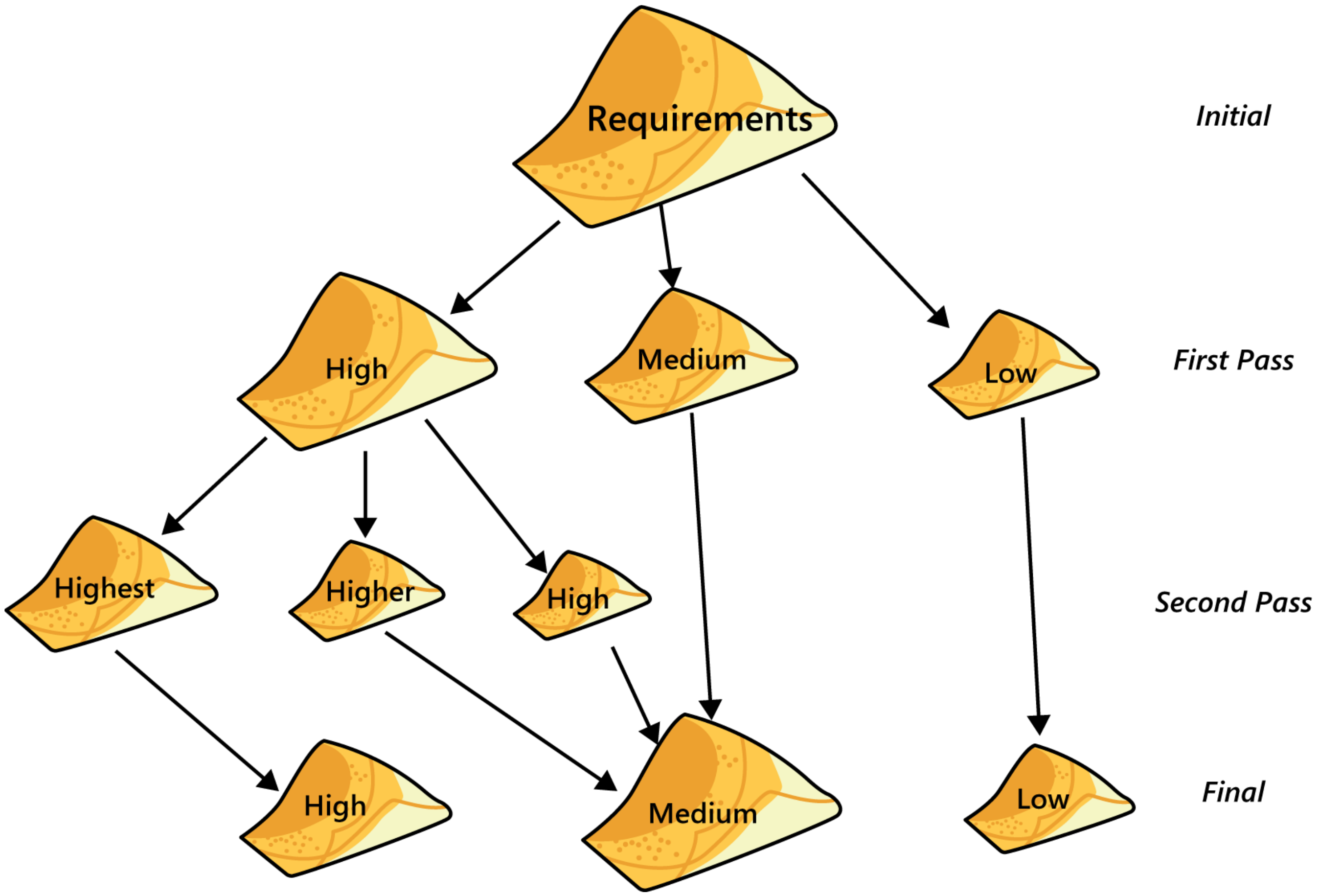
QS STARS RATED FOR EXCELLENCE 2012

FPT Education

FPT UNIVERSITY

Some prioritization techniques

- Some techniques are:
  - Three-level scale: The way to use this technique



Some prioritization techniques

- Some techniques are:
  - MoSCoW: Must – Should – Could – Won't
    - **M**ust: The requirement must be satisfied for the solution to be considered a success.
    - **S**hould: The requirement is important and should be included in the solution if possible, but it's not mandatory to success.
    - **C**ould: It's a desirable capability, but one that could be deferred or eliminated. Implement it only if time and resources permit.
    - **W**on't: This indicates a requirement that will not be implemented at this time but could be included in a future release.
  - $100: One way to make prioritization more tangible is to cast it in terms of an actual resource: money.
    - Give the prioritization team 100 imaginary dollars to work with.
    - Team members allocate these dollars to "buy" items that they would like to have implemented from the complete set of candidate requirements.
    - Then add up the total number of dollars assigned to each requirement to see which ones collectively come out as having the highest priority.

Prioritization based on value, cost, and risk

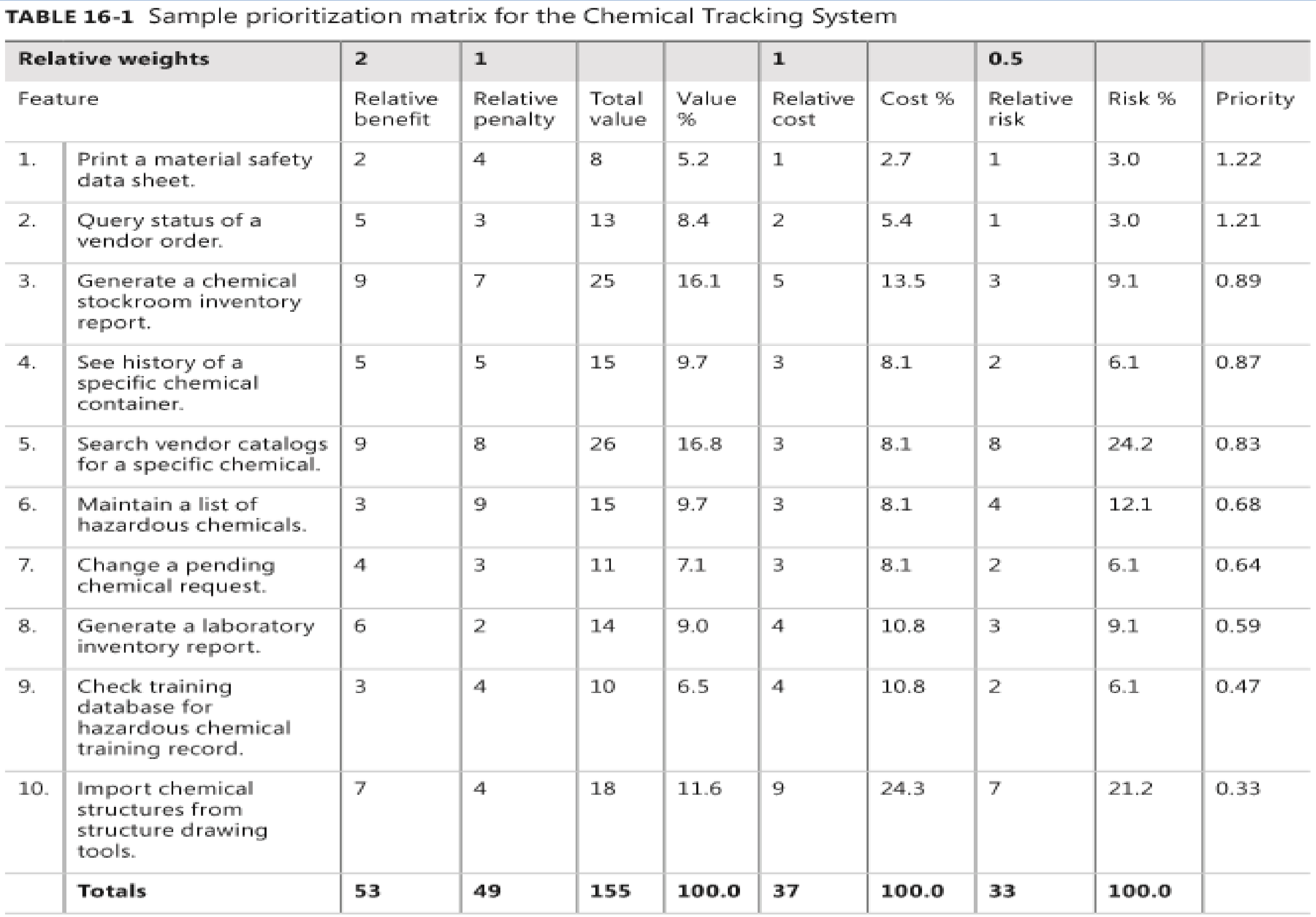


TABLE 16-1 Sample prioritization matrix for the Chemical Tracking System

Relative weights		**2**	**1**			**1**		**0.5**		
Feature		Relative benefit	Relative penalty	Total value	Value %	Relative cost	Cost %	Relative risk	Risk %	Priority
1.	Print a material safety data sheet.	2	4	8	5.2	1	2.7	1	3.0	1.22
2.	Query status of a vendor order.	5	3	13	8.4	2	5.4	1	3.0	1.21
3.	Generate a chemical stockroom inventory report.	9	7	25	16.1	5	13.5	3	9.1	0.89
4.	See history of a specific chemical container.	5	5	15	9.7	3	8.1	2	6.1	0.87
5.	Search vendor catalogs for a specific chemical.	9	8	26	16.8	3	8.1	8	24.2	0.83
6.	Maintain a list of hazardous chemicals.	3	9	15	9.7	3	8.1	4	12.1	0.68
7.	Change a pending chemical request.	4	3	11	7.1	3	8.1	2	6.1	0.64
8.	Generate a laboratory inventory report.	6	2	14	9.0	4	10.8	3	9.1	0.59
9.	Check training database for hazardous chemical training record.	3	4	10	6.5	4	10.8	2	6.1	0.47
10.	Import chemical structures from structure drawing tools.	7	4	18	11.6	9	24.3	7	21.2	0.33
	Totals	**53**	**49**	**155**	**100.0**	**37**	**100.0**	**33**	**100.0**	

Prioritization based on value, cost, and risk

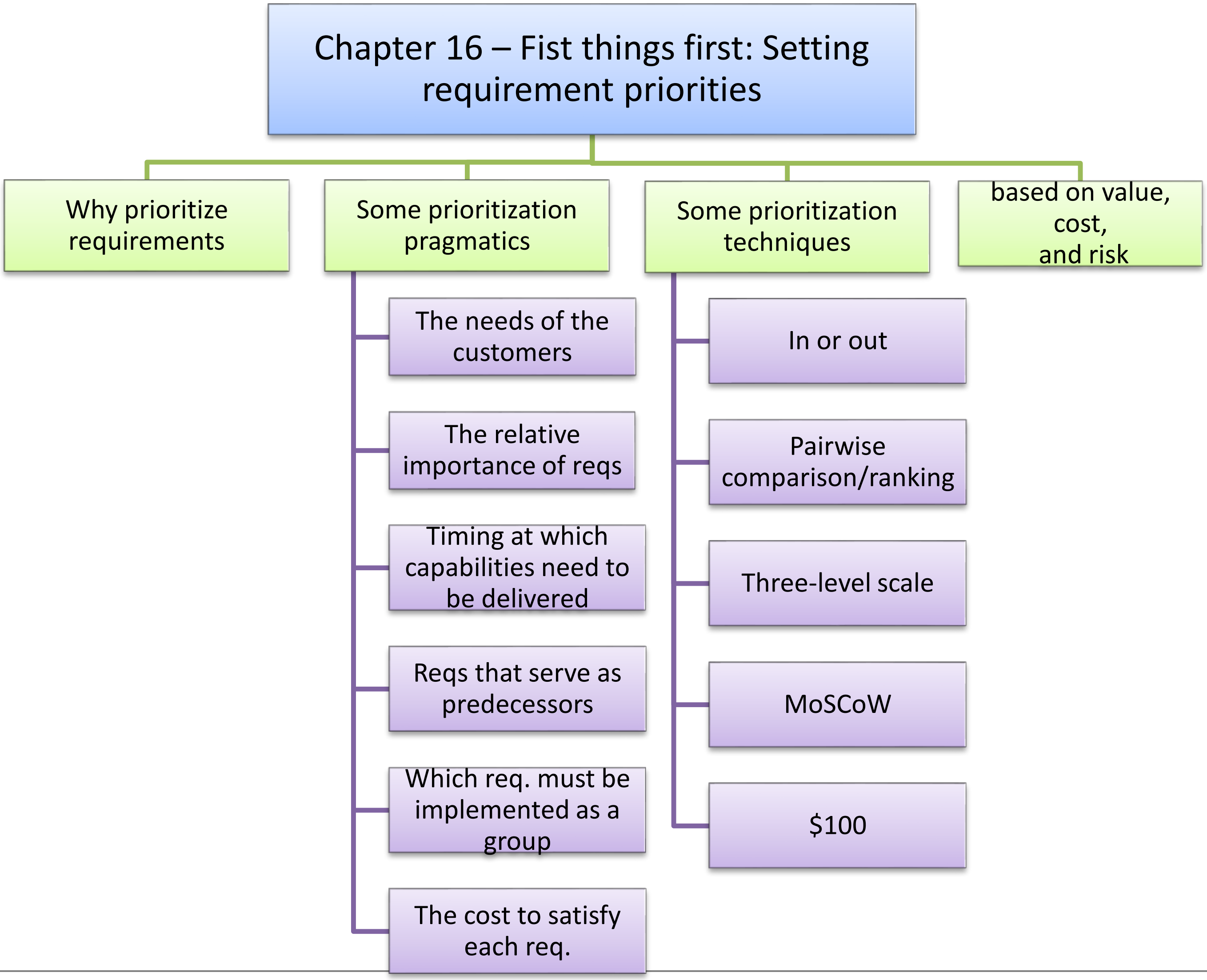
1. List in the spreadsheet all the features, use cases, use case flows, user stories, or functional requirements that you want to prioritize against each other.
2. The customer representatives estimate the relative benefit each feature would provide to the customer or to the business on a scale of 1 to 9.
3. Estimate the relative penalty that the customer or the business would suffer if each feature were *not* included. Again, use a scale of 1 to 9. A rating of 1 means that no one will be upset if it's absent; 9 indicates a serious downside.
4. The spreadsheet calculates the total value for each feature as the sum of its benefit and penalty scores (weighted as described later in the chapter).
5. Developers estimate the relative cost of implementing each feature, again on a scale of 1 (quick and easy) to 9 (time-consuming and expensive). The spreadsheet will calculate the percentage of the total cost that each feature contributes.
6. Developers rate the relative technical (not business) risk associated with each feature on a scale of 1 to 9. Technical risk is the probability of *not* getting the feature right on the first try.
7. The priority value for each feature by using the following formula:

   (value %)/(cost % + risk %)
8. Finally, sort the list of features in descending order by calculated priority, the rightmost column.

FPT Education

FPT UNIVERSITY

Review chapter 16


Chapter 16 – Fist things first: Setting requirement priorities
Why prioritize requirements
Some prioritization pragmatics
Some prioritization techniques
based on value, cost, and risk
The needs of the customers
The relative importance of reqs
Timing at which capabilities need to be delivered
Reqs that serve as predecessors
Which req. must be implemented as a group
The cost to satisfy each req.
In or out
Pairwise comparison/ranking
Three-level scale
MoSCoW
$100

THE END

THANK YOU!