



Requirements

CSCI-4448 - Boese



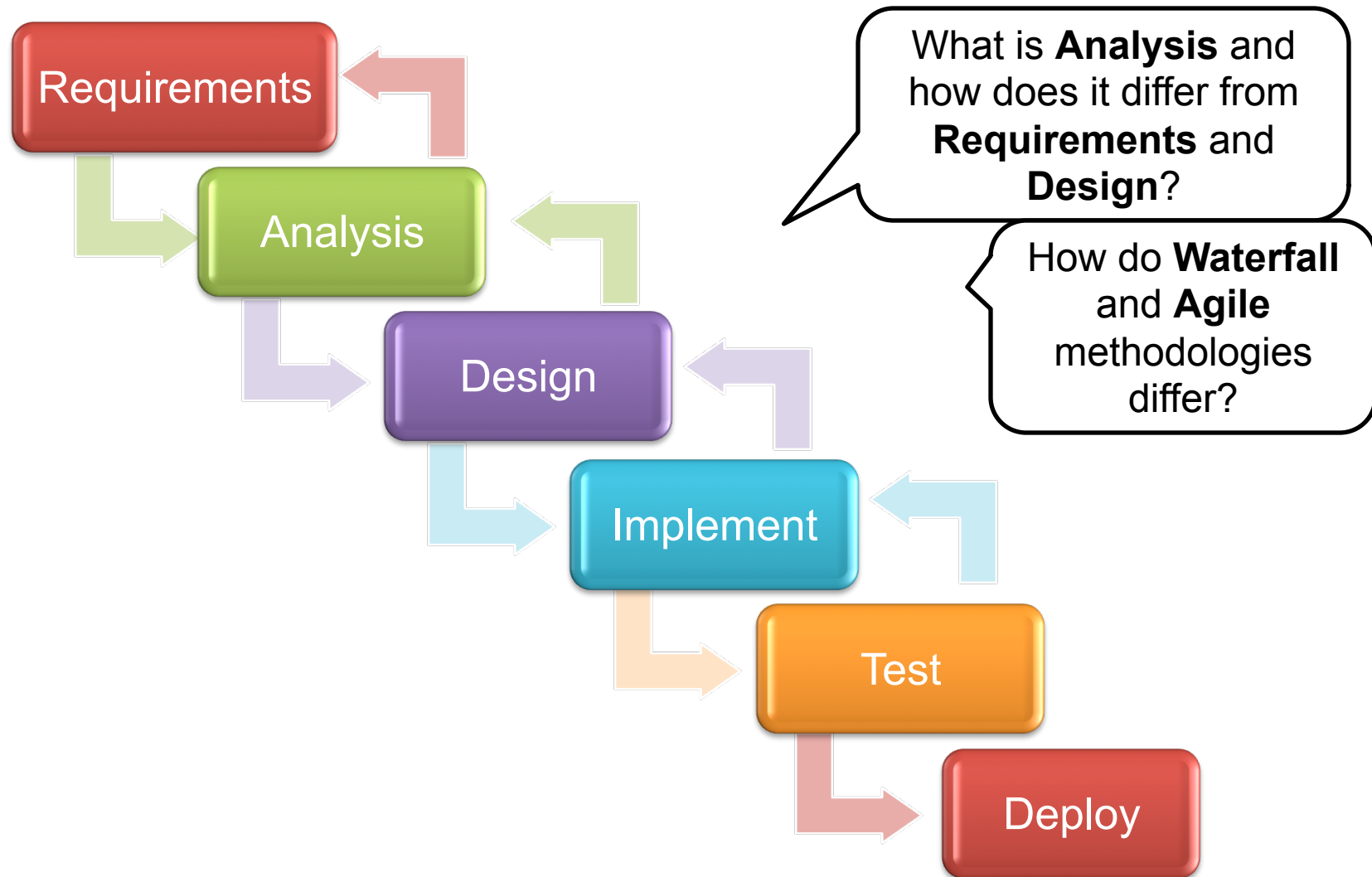
University of Colorado **Boulder**

Objectives

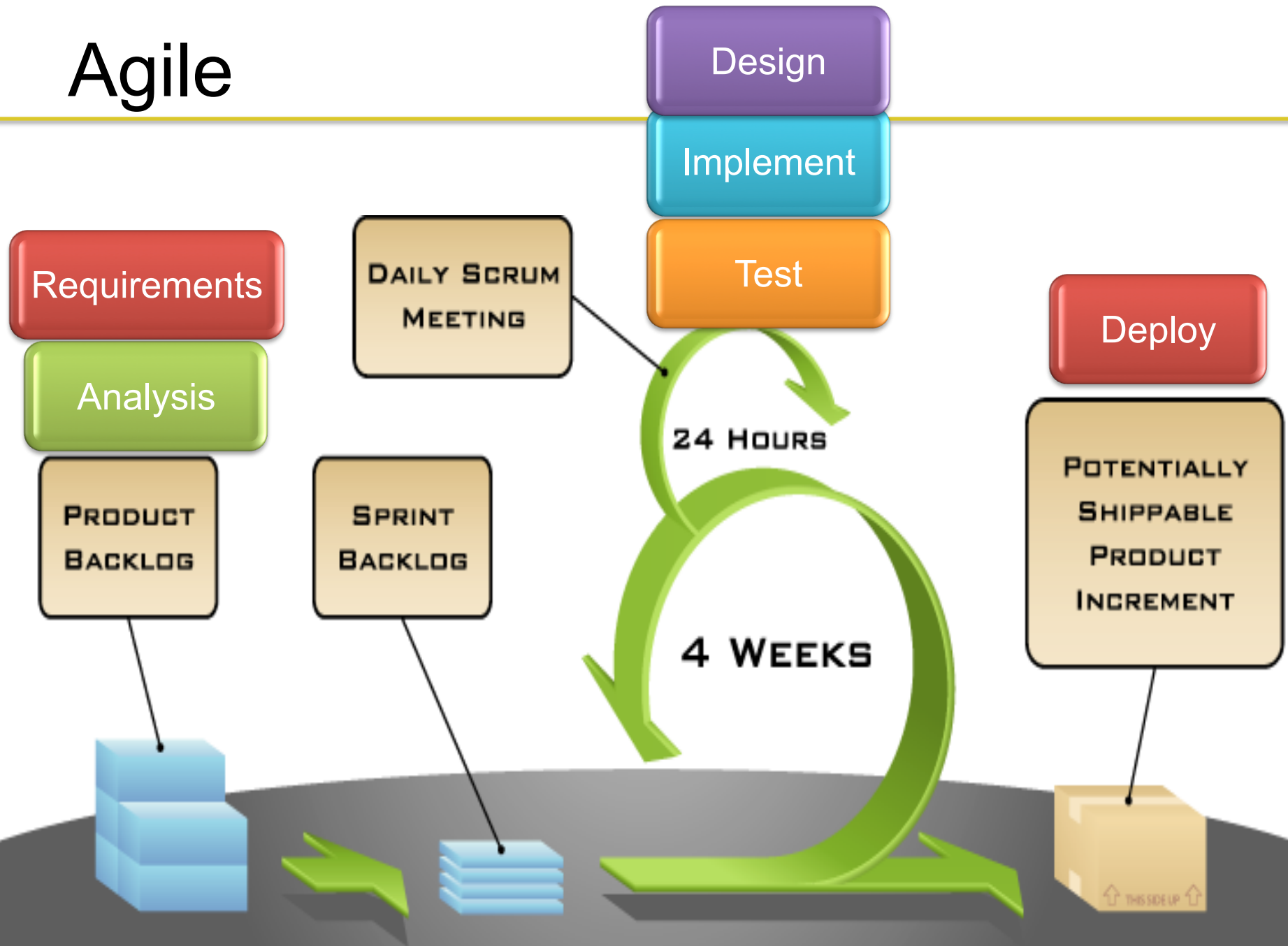
- Understand an overview of SDLC
- Understand how to determine and analyze requirements
- Know how/when to apply Unified Modeling Language (UML)

Overview SDLC

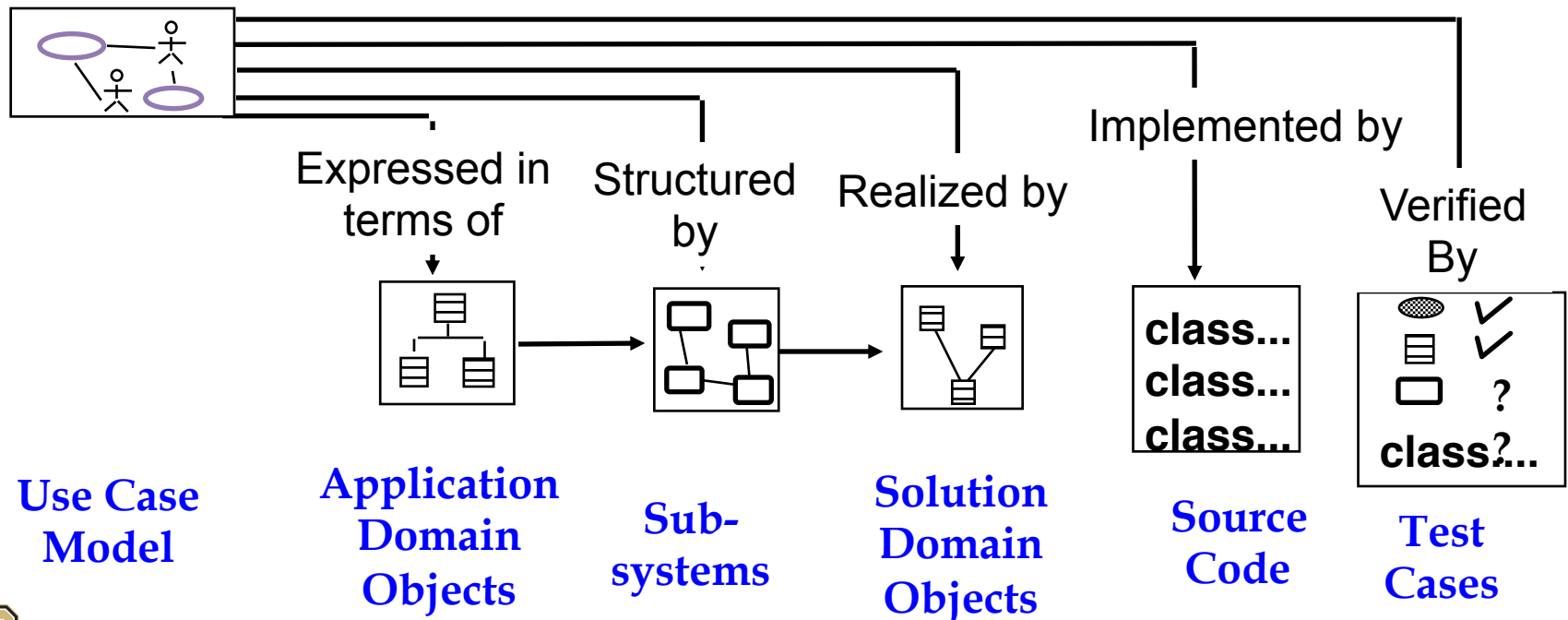
SDLC



Agile



Software Lifecycle Activities ...and their models

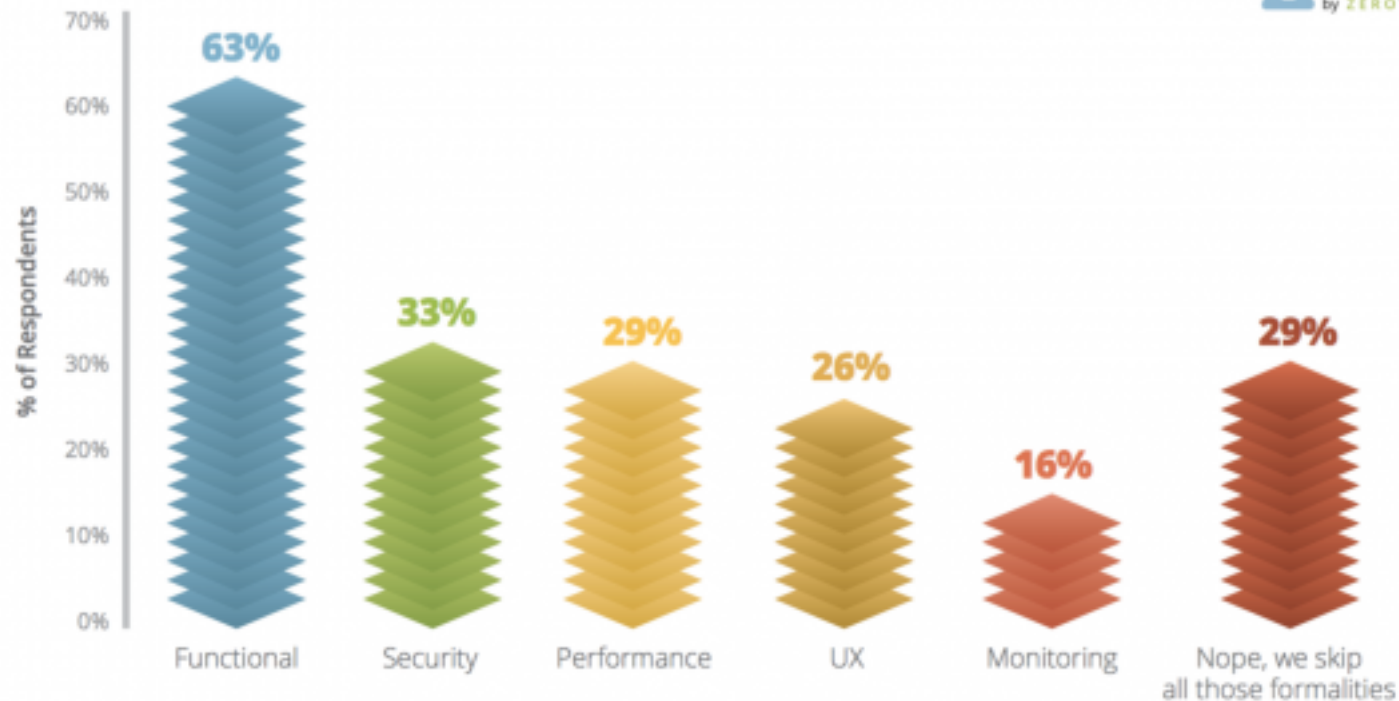


Determining Requirements



Formal Requirements

Figure 1.24 Use of Formal Requirements by Project Area



Determining Requirements

- With client
- List
- Client will tell you
 - What they *want*
 - What they *think* they need
- Your job
 - Determine what they *actually* need

Another Perspective

From Steve Jobs:

"Some people say, 'Give the customers what they want.'"

But that's not my approach.

Our job is to figure out what they're going to want before they do.

I think Henry Ford once said, 'If I'd asked customers what they wanted, they would have told me, 'A faster horse!'"

*People don't know what they want until you show it to them.
That's why I never rely on market research.
Our task is to read things that are not yet on the page.*



How the customer explained it



How the Project Leader understood it



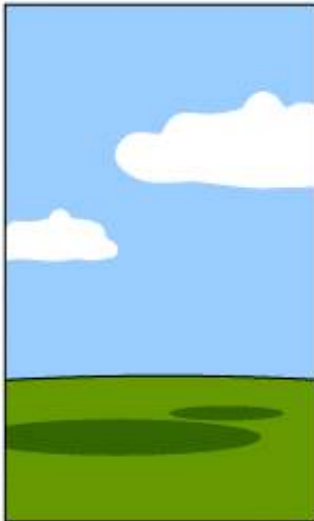
How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



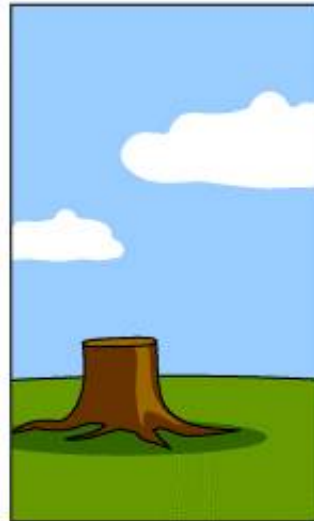
How the project was documented



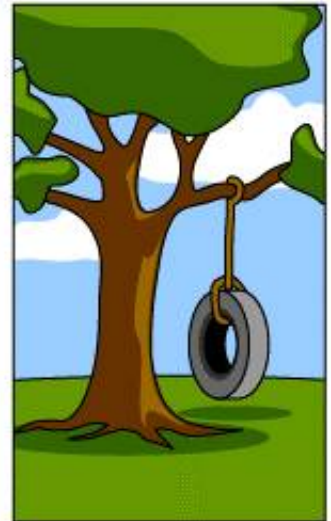
What operations installed



How the customer was billed



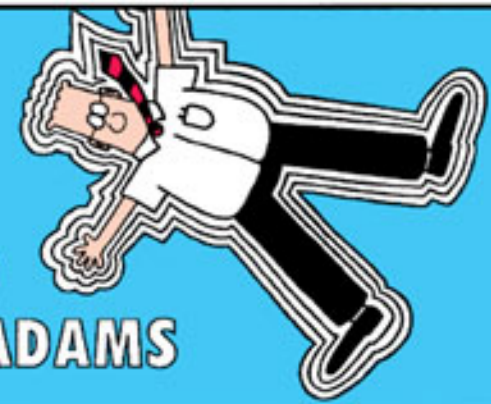
How it was supported



What the customer really needed



DILBERT®



BY
SCOTT ADAMS



E-mail: SCOTTADAMS@AOL.COM



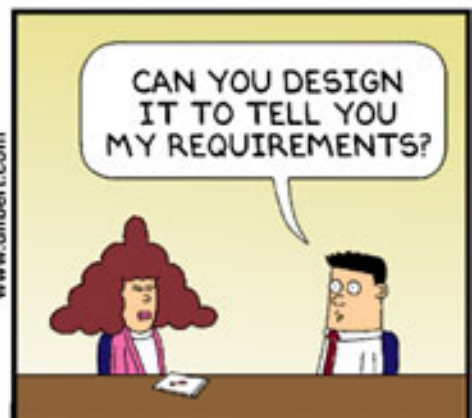
© 2006 Scott Adams, Inc./Dist. by UFS, Inc.



1-7-98



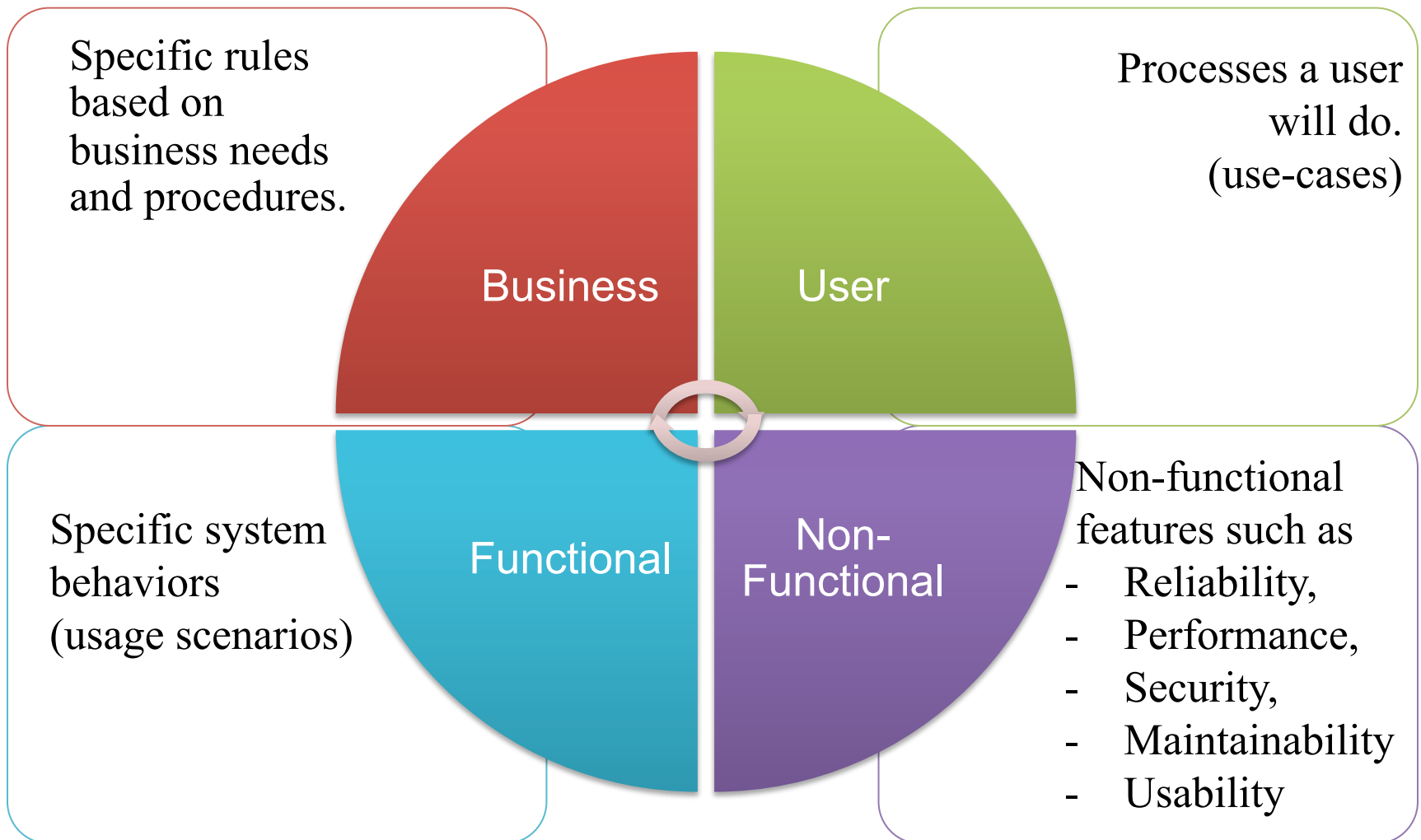
www.dilbert.com



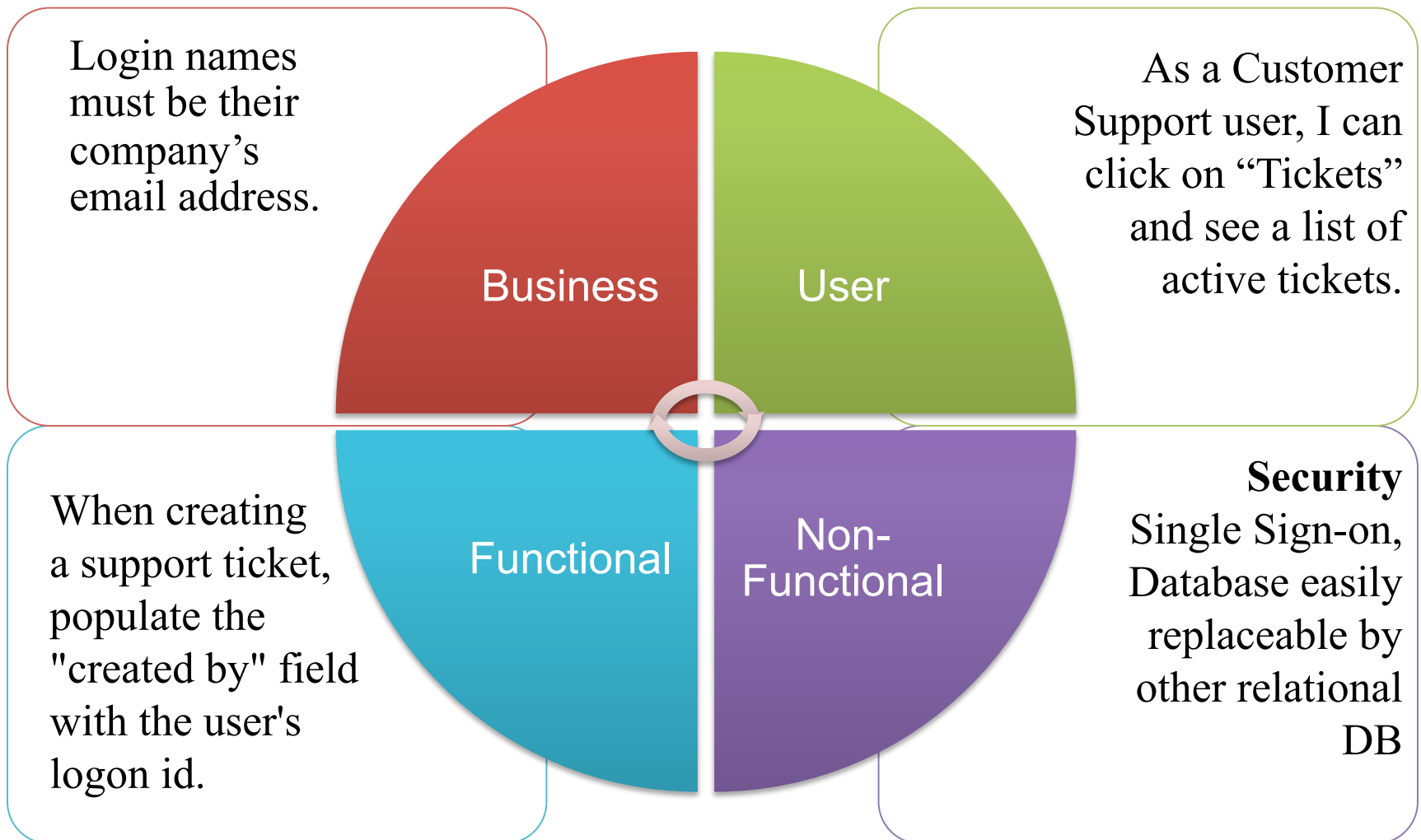
Determining Requirements

- Focus on listing out the client's needs
 - Spreadsheet
 - Document
 - Drawings/sketches
 - Diagrams
- As long as it is clearly specified
 - *These requirements are used at the end of the project to designate whether you fulfilled the contract! Also used for User Acceptance Tests*
 - *Danger: ambiguous requirements that can be interpreted multiple ways!*
- Add ID number to each requirement for reference
 - Useful for ensuring test cases cover the requirements
 - Referencing during meetings

Types of Requirements



Types of Requirements - Examples



Requirements Examples

User Requirements		
ID	Description	Priority
US-01	As a system admin, I want to see a dashboard of weekly usage by user type so that I can monitor the usefulness of the system.	Medium
US-03	As a customer service rep, I want to be notified of failures so that I can respond to all users within an hour.	High

Non-Functional Requirements		
ID	Description	Priority
NFR-01	<u>Performance</u> : Upon providing correct credentials, user shall be directed to the landing page in under 7 seconds.	Medium
NFR-03	<u>Platform Constraints</u> : Login functionality shall behave the same on different platforms (Linux/Windows/iOS/mobile).	High

Non-Functional Requirements

Category	Example questions
Usability	<ul style="list-style-type: none">• What is the level of expertise of the user?• What user interface standards are familiar to the user?• What documentation should be provided to the user?
Reliability <i>(including robustness, safety, and security)</i>	<ul style="list-style-type: none">• How reliable, available, and robust should the system be?• Is restarting the system acceptable in the event of a failure?• How much data can the system lose?• How should the system handle exceptions?• Are there safety requirements of the system?• Are there security requirements of the system?
Performance	<ul style="list-style-type: none">• How responsive should the system be?• Are any user tasks time critical?• How many concurrent users should it support?• How large is a typical data store for comparable systems?• What is the worst latency that is acceptable to users?
Supportability <i>(including maintainability and portability)</i>	<ul style="list-style-type: none">• What are the foreseen extensions to the system?• Who maintains the system?• Are there plans to port the system to different software or hardware environments?



Non-Functional Requirements

Implementation	<ul style="list-style-type: none">• Are there constraints on the hardware platform?• Are constraints imposed by the maintenance team?• Are constraints imposed by the testing team?
Interface	<ul style="list-style-type: none">• Should the system interact with any existing systems?• How are data exported/imported into the system?• What standards in use by the client should be supported by the system?
Operation	<ul style="list-style-type: none">• Who manages the running system?
Packaging	<ul style="list-style-type: none">• Who installs the system?• How many installations are foreseen?• Are there time constraints on the installation?
Legal	<ul style="list-style-type: none">• How should the system be licensed?• Are any liability issues associated with system failures?• Are any royalties or licensing fees incurred by using specific algorithms or components?

Agile Requirements

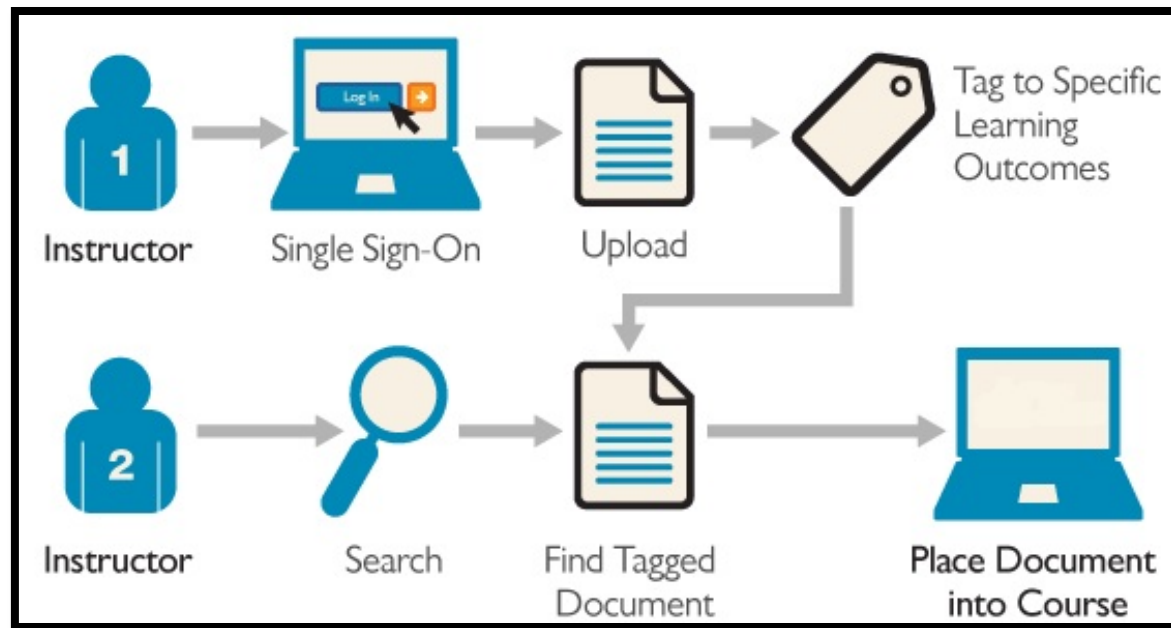


Why this format?

- **As a** <user>,
 I need <a task>
 so that I <can accomplish a goal>.
- **As a** user, **I need to** indicate folders not to backup **so that** my backup drive isn't filled up with things I don't need saved.
- **As an** administrator, **I want** the login usernames to be the same as their company email address **so that** we can prevent spam accounts.

Vision

- Once you have the client's vision, a “best-practice” is to write the vision (statement or diagram) at the top of all documents/presentations...
- If the vision changes, that is a BIG DEAL and can void the contract *(or increment the cost for the contract for new features!)*



Requirements Analysis

- Are the requirements feasible?
 - Time, budget, resources
- Specify with diagrams
 - *(diagram the requirements, not the design. Diagramming the design comes later)*
- Why do analysis?
 - Fully **understand** the requirements
 - Identify **gaps** in the requirements
 - Identify **errors** and **conflicts** in requirements
 - Use to **verify completion** of project/contract

*** Very important to avoid issues over contract/money!

Relative cost to fix a fault that could have been fixed in requirements

