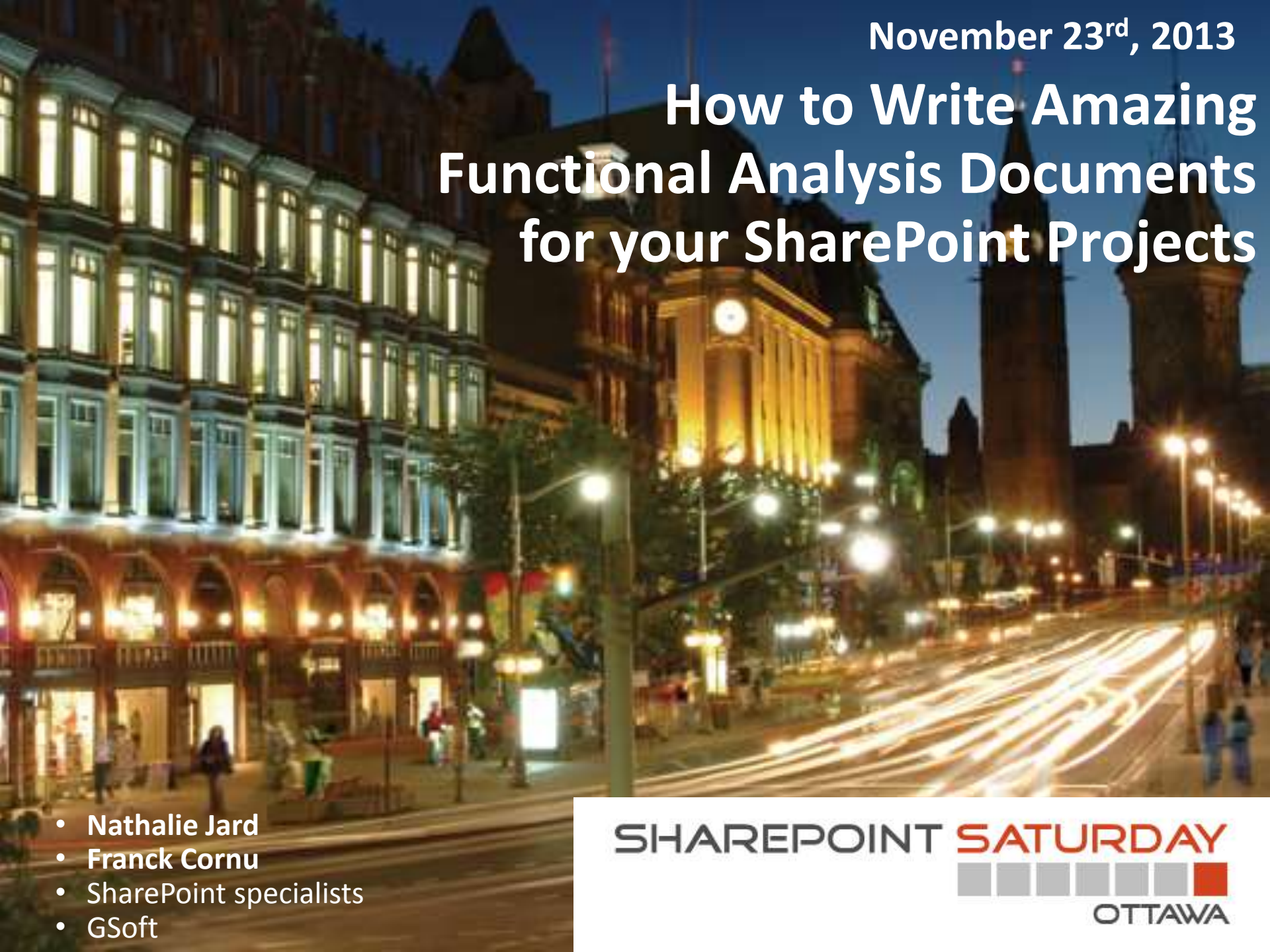


November 23rd, 2013

# How to Write Amazing Functional Analysis Documents for your SharePoint Projects


- **Nathalie Jard**
- **Franck Cornu**
- SharePoint specialists
- GSoft


SHAREPOINT SATURDAY OTTAWA

SHAREPOINT SATURDAY OTTAWA

# Thank you to all of our Sponsors!!















Who

# ARE WE?

## Nathalie Jard

- Scrum Master
- SharePoint Analyst

nathalie.jard@gsoft.com

@najard

## Franck Cornu

- SharePoint Analyst & Developer

franck.cornu@gsoft.com

@franckcornu

Dynamite

http://www.gsoft.com/fr/blogue?categorie=sharepoint

SESSION

# Plan

## A Requirements definition

1 - Define requirements

2 - Prioritize requirements

## B Functional analysis

1 - Prerequisite

2 – 9 steps method

1. Visualize the requirement
2. Describe information
3. Define relationships between data types
4. Effectively storing information
5. Determine the data flow directions
6. Define data access point
7. Define criteria for data recovery
8. Define relational behaviors
9. Define the information display

# Part A

## Requirements definition

**Focus first on the "What"?**

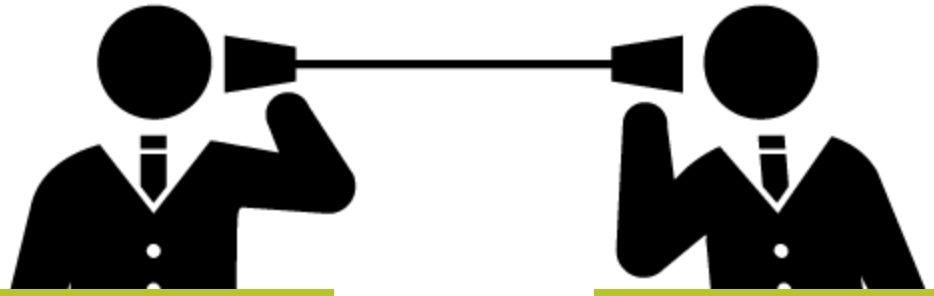
*"The value of an idea lies in the using of it."*

- Thomas A. Edison

1

## Define requirements

# Do you know the difference between?



**Business analyst** And **Functional analyst**

Part A

- Understand business process
  - Read documentation if exists
  - Identify super users or product owners
  - Identify actual problems on process or current solution
  - Animate workshops

Part B

- Understand SharePoint mechanisms
  - Knows possibilities with SharePoint and transcribe them to functional questions.
  - Link between IT and business
  - Product owner(s) best friend

1

## Define requirements

# By writing a backlog


- Very first document
  - Identify what will be done in your project
  - **Don’t neglect it**
- Used with agile methodologies
  - Sprintable and releasable (ex **SCRUM**)
  - Prioritized
  - Flexible
  - Requirements as stories
    - **Independent analysis on each**
- This document can be
  - A simple flat list
    - Avoid TFS…
  - A map (?!?)

## Define requirements – Tools & Techniques

# User Story Mapping

## Definition

- Organize and prioritize user stories in a backlog(*)
  - make **visible** the workflow or value chain
  - show the **relationships** of larger stories to their child stories
  - help confirm the **completeness** of your backlog
  - provide a useful **context** for prioritization
  - plan releases in complete and **valuable slices** of functionality.
- Workshops (maximum 4 peoples)
  - Identity product owners per functional domains
  - Tools
    - Coloured post-it and whiteboard
    - Electronic format (SpecLog $)



*http://www.agileproductdesign.com

## Define requirements – Tools & Techniques

# User Story Mapping

## Concept

**Impacts**

**Is there any constraints (IT, functional, human..) on?**

- **Business goals** — **Why you do it?**
- **Actors/Profiles** — **Who will use it?**
- **Activities** — **What are their responsibilities on it?**

---

- **User stories** — **What are their tasks within those responsibilities?**

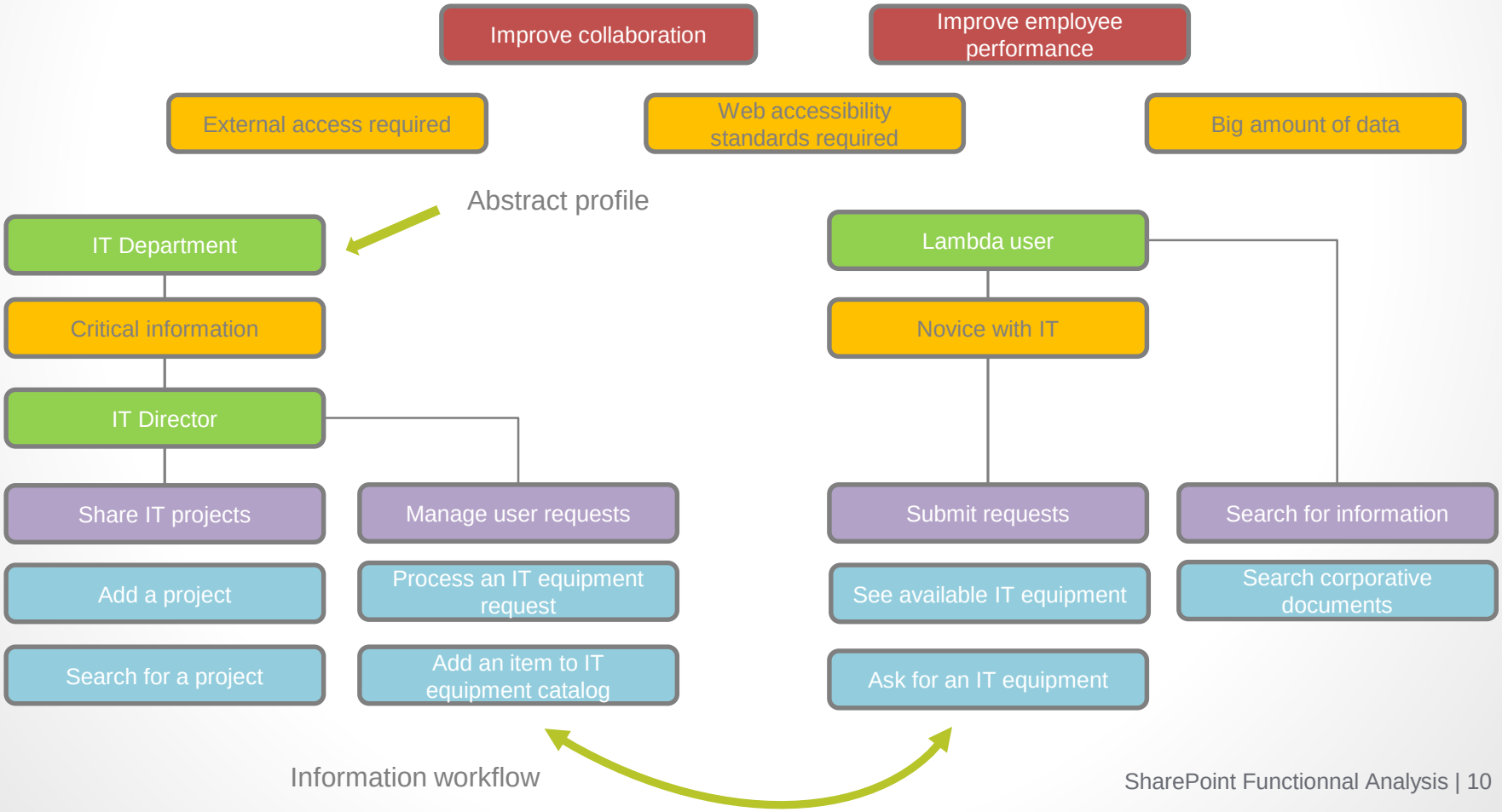
Business priority: + / –

User workflow

## Define requirements – Tools & Techniques

# User Story Mapping

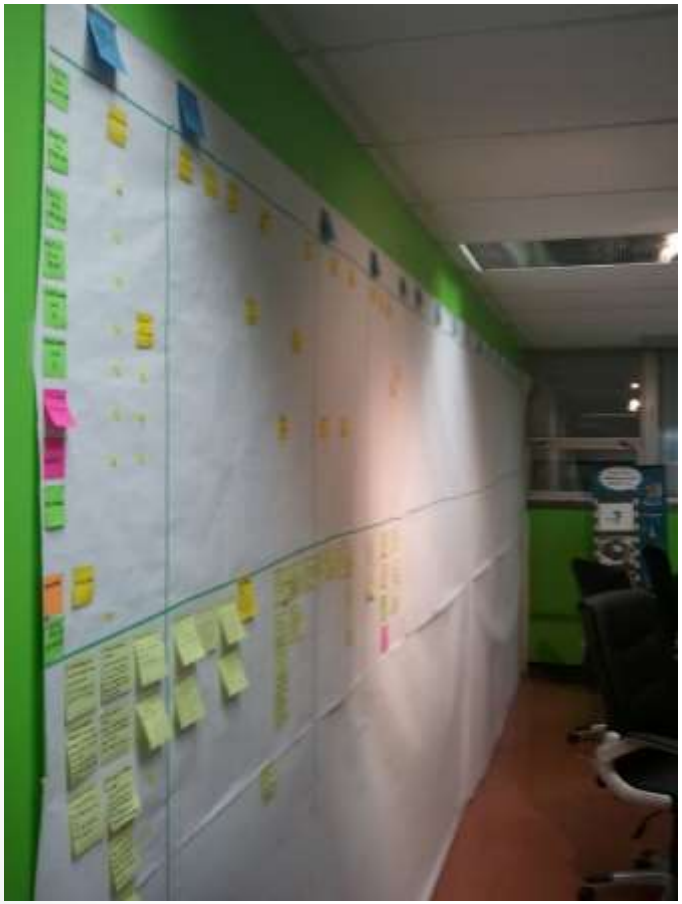
## Example

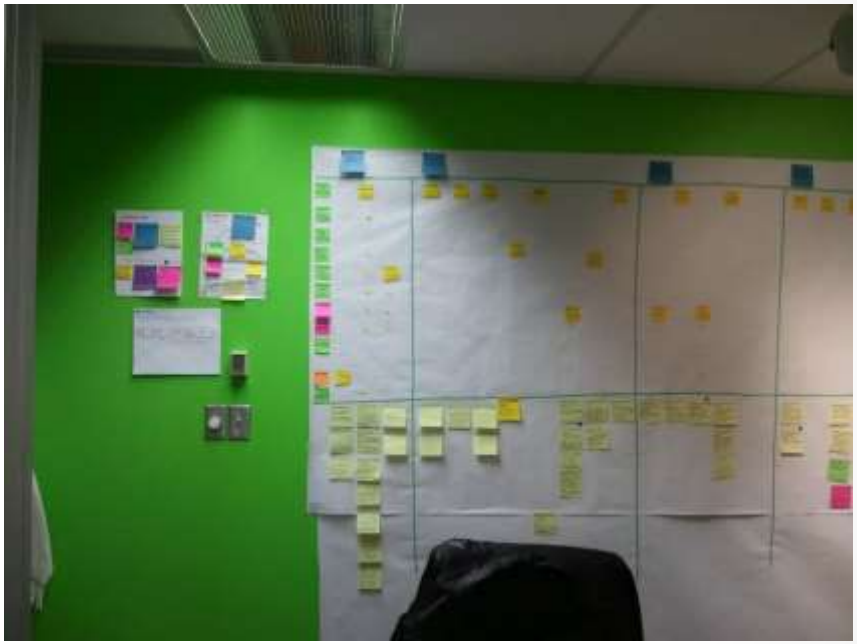


## Define requirements – Tools & Techniques

# User Story Mapping

### Example in real life





**PS: Not mandatory to use a full wall ☺**

## Define requirements – Tools & Techniques

# User Story Mapping

## Summary

### ✓ Live in a dream, don't talk about SharePoint capabilities

Everything is possible here !

### ✓ Focus on roles and responsabilites

Who can do what in the system? **This**, is governance.

### ✓ Focus on requirements, not solutions

No more: « I want a search engine » or « I want a SharePoint list with five columns »

### ✓ Reveal information flows

That they will help you in the future to design your information architecture

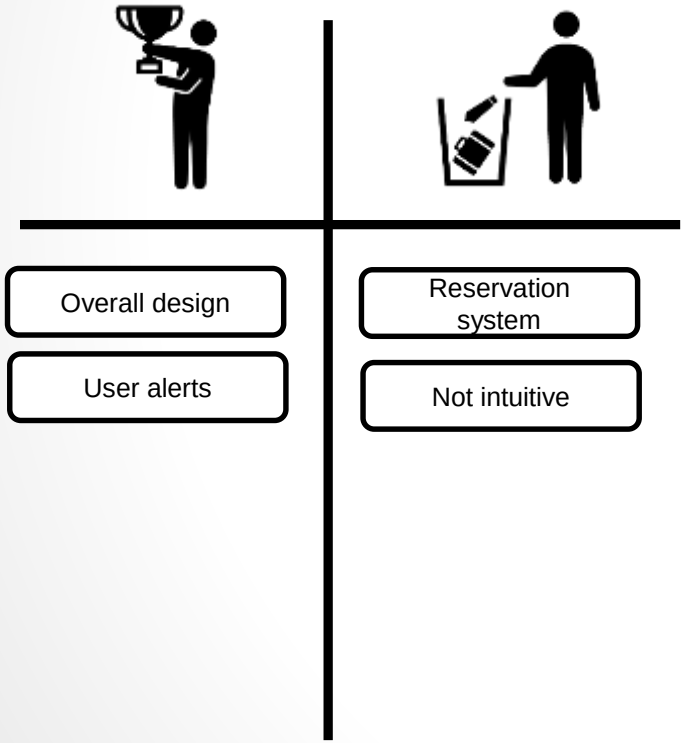
### ✓ Super effective but sometimes hard to assimilate

Practice, practice and practice. Dont be afraid to try. All you need is listening and rigor.

## Define requirements – Tools & Techniques

# Keep/Don't keep

## Definition



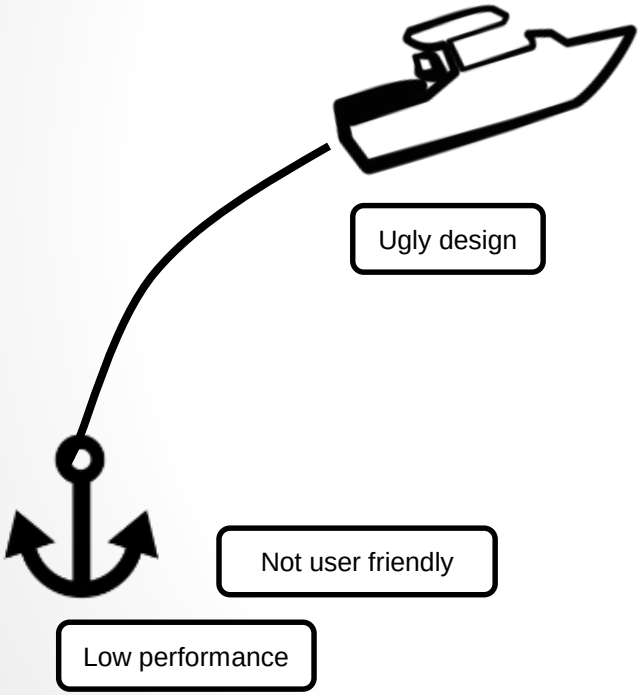

- Identify what users like or don't like in their actual system or process
  - particularly suitable for a system upgrade/revision project
  - result as impacts in your map (all levels)
- Tools
  - pencils and whiteboard
  - post-it
  - focus groups by functional domain

*Source: http://innovationgames.com*

# Define requirements – Tools & Techniques

## Speed boat

### Definition




- Identify what slows down user efficiency
  - Most problems lie deeper
  - Helps you to identify impact **priorities**
- Tools
  - pencils and whiteboard
  - post-it
  - focus groups by functional domain

*Source: http://innovationgames.com*

2

## Prioritize requirements

# Why prioritize?

- You have to start with something
  - « I want it all » is not a valid response
- Agile development guideline
  - First deliver features that give you the most value

*Source: http://innovationgames.com*

2

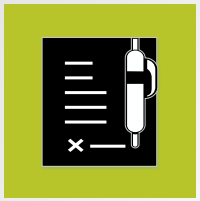
## Prioritize requirements – Tools & Techniques

# Buy a feature

## Definition

- Distribute your fake money on available requirements
  - Identify priorities
- Tools
  - fake money
  - product backlog
  - timebox

*Source: http://innovationgames.com*





# Summary

- Prioritized business requirements list (MoSCoW)

| Priority | User Story |
|---|---|
| **Priority 0 – Must have** | Add a project |
| | Search a project |
| | Search corporative documents |
| **Priority 1 – Should have** | Process an IT equipment process |
| | See available IT equipment |
| | Ask for an IT equipment |
| **Priority 2 – Nice to have** | Add an item to IT equipment catalog |

# Part B

## Functional analysis

**Focus next on the ‘How"?**

*"If you can't explain it simply, you don't understand it well enough."*

- Albert Einstein

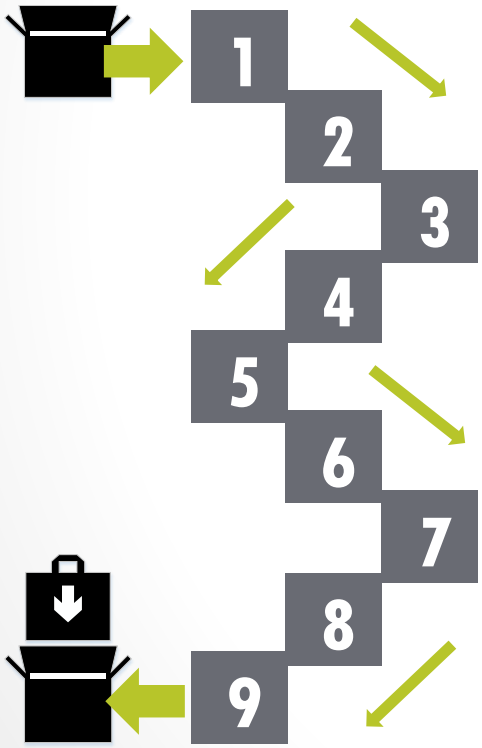
B

## Functional analysis

# With SharePoint

## Methodology

AVAILABLE ON SKYDRIVE




- 9 simple steps
  - Based on functional questions
  - SharePoint OOTB oriented
- Applies on a **single story**
  - Building little autonomous subsystem with its own functional analysis
- All steps are not mandatory
  - Depends on your user story
    - « Read » oriented VS « Write » oriented

B

Functional analysis

# With SharePoint

## Top SharePoint analysis facts


### 1 Analy…what??

SharePoint is like Legos®, you can do anything, for better for worse. It's up to you to choose the right brick for the right use

### 2 The SharePoint trap: when the means condition the needs

Tell me what SharePoint can do, I will tell you what you need…

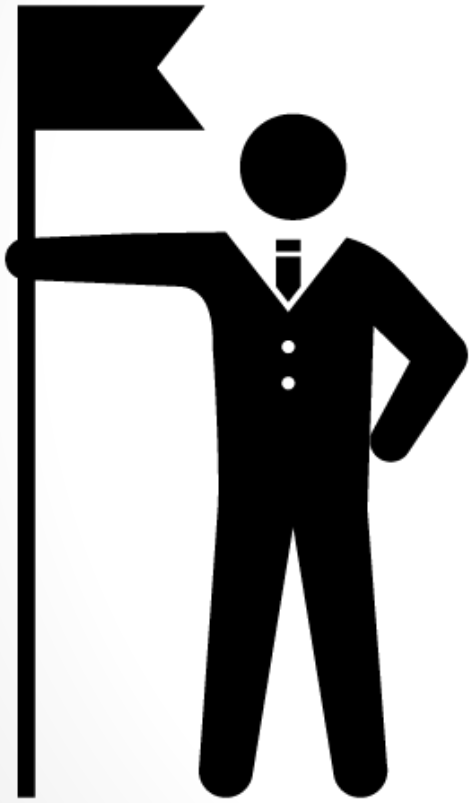
### 3 Solutions are not close enough to users day to day needs

IT department is not often the best resource to determine whole user requirements…

B

Methodology summary

# Your functional analysis with SharePoint

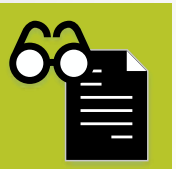


1. Visualize the requirement
2. Describe information
3. Define relationships between data types
4. Effectively storing information
5. Determine the data flow directions
6. Define data access point
7. Define criteria for data recovery
8. Define relational behaviors
9. Define the information display

0

Prerequisite

# How to access data in SharePoint ?



## Structured Query Language VS Search language

### Benefits

- Represent the SharePoint database reality at the moment of the query.
- Retrieves linked items quite easily.

### Disadvantages

- Requires defining each field involved in the query.

### Suitable for

- Access to targeted metadata
- Access to content subject to continuous changes

### Benefits

- No need to know the field names in which to look for.
- Easy query building

### Disadvantages

- Represents the search index reality and not the SharePoint content database one.

### Suitable for

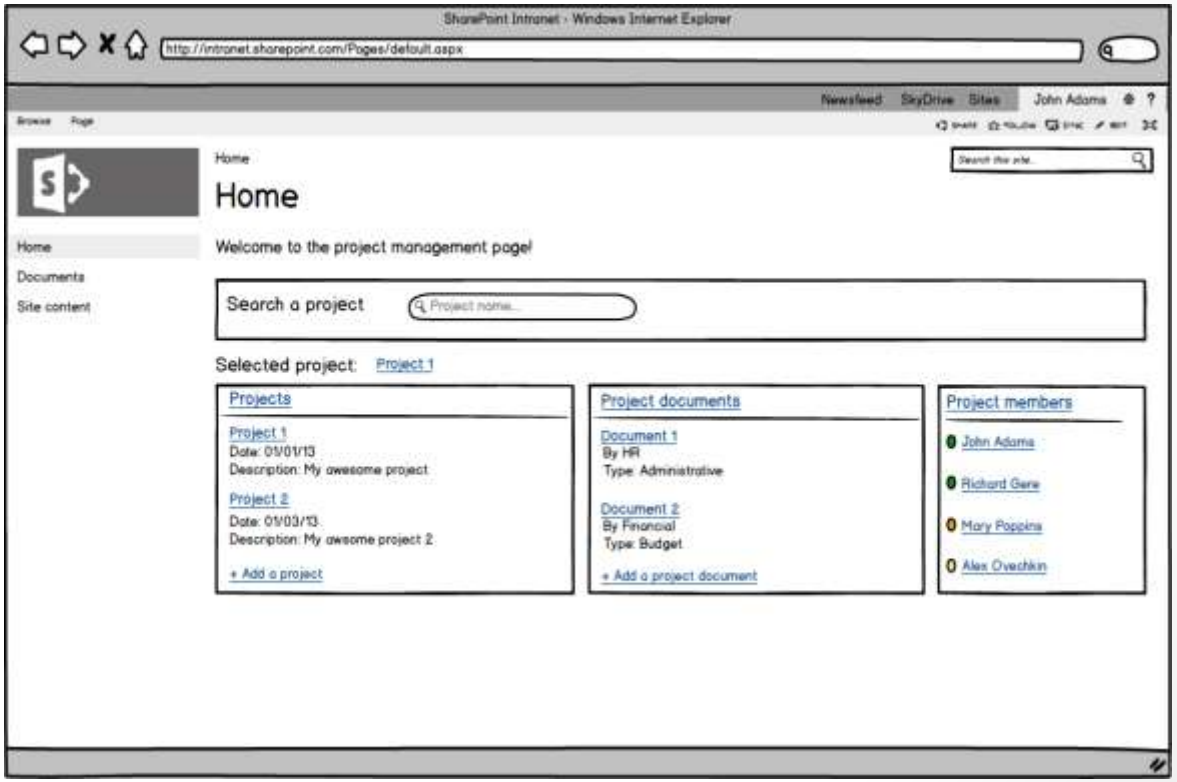
- Keywords based search in "Free Text" mode
- Access to relatively static content.

1

## Visualizing the requirement

# What does the requirement look like?

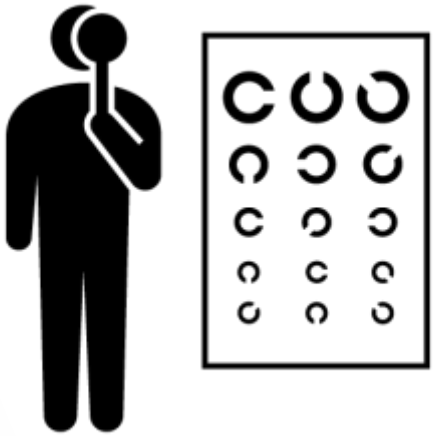
## Wireframe(s)



2

## Describe information

# What are the information types present in the requirement?



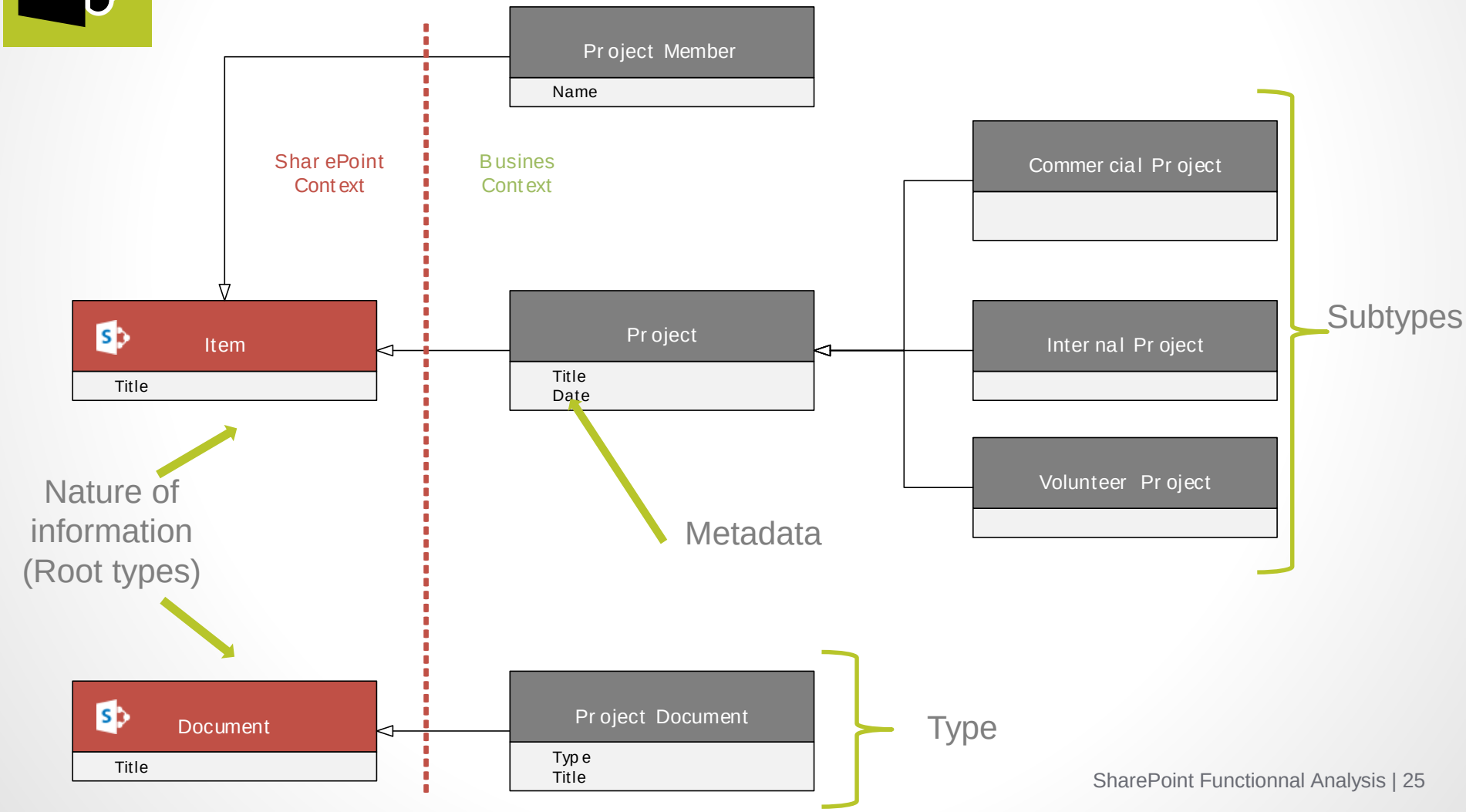
- Static or dynamic information?
  - Reusability concerns
- Types hierarchy?
  - Subtypes, specializations
- Metadata?
  - Visible **and** hidden

2

## Describe information

# With SharePoint

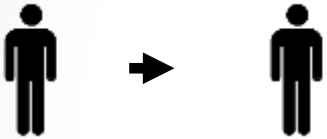
## Content Types and Columns



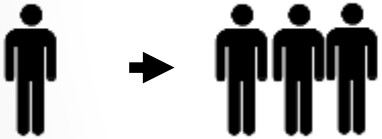
3

## Define relationships between data types

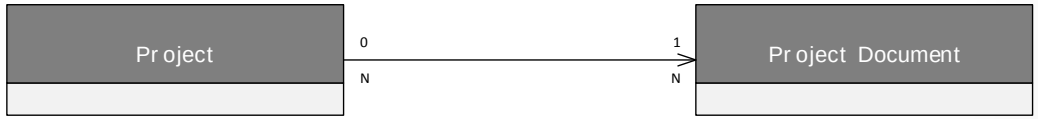
# What are the relationships between these data items?



- Relationships possibilities
  - One to one

  - One to many




  - Many to many

3

## Define relationships between data types

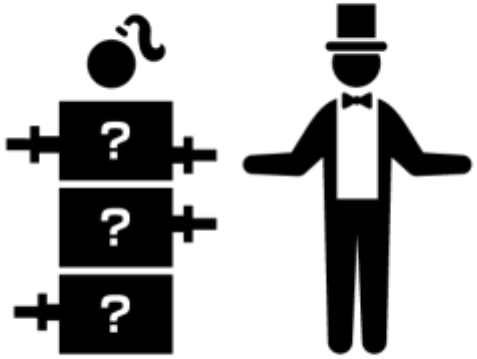
# With SharePoint

## Possibilities

| | 1 ▶ 1 | 1 ▶ N | N ▶ N |
|---|---|---|---|
| Lookup Fields | ✓ | ✓ | ✓ |
| Managed Metadata | ✓ | ✓ | ✓ |
| Documents Sets | ✓ | ✓ | |
| Folders | ✓ | ✓ | |

## Effectively storing information

# How should data be stored within a requirement?



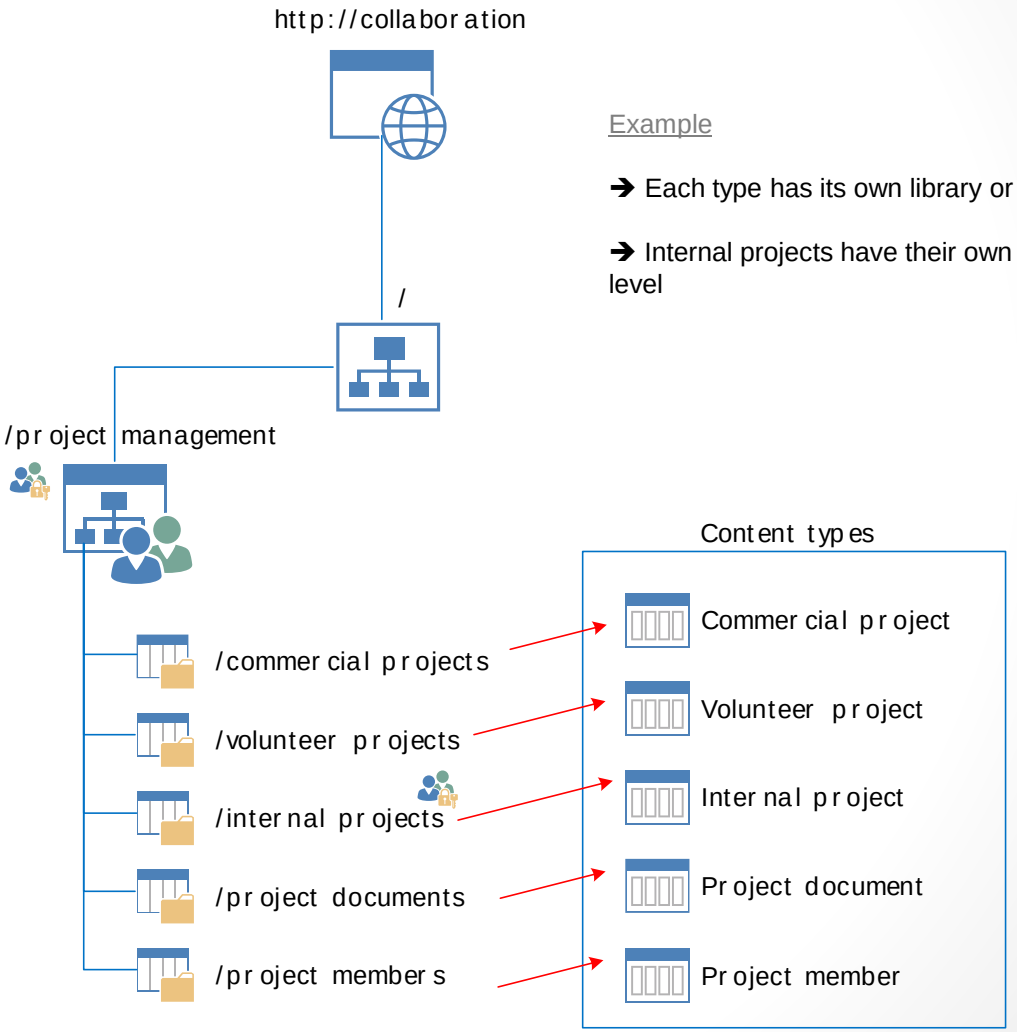

- Data distribution criteria
  - Volume
    - What is the amount of data?
  - Information security
    - Who need to access these data?
  - Business context
    - Does the structure have to follow a business hierarchy?

4

## Effectively storing information

# With SharePoint

- Web applications
- Sites collections
- Sites
- Lists and libraries
- List items




Example

➔ Each type has its own library or list

➔ Internal projects have their own permission level

5

## Determining the data flow directions

# Within my SharePoint data structure, how does the data flow?

- Output flow (data read)
  - The system displays information to the user
- Input flow (data write)
  - Users add or edit data through the system components

5

## Determining the data flow directions

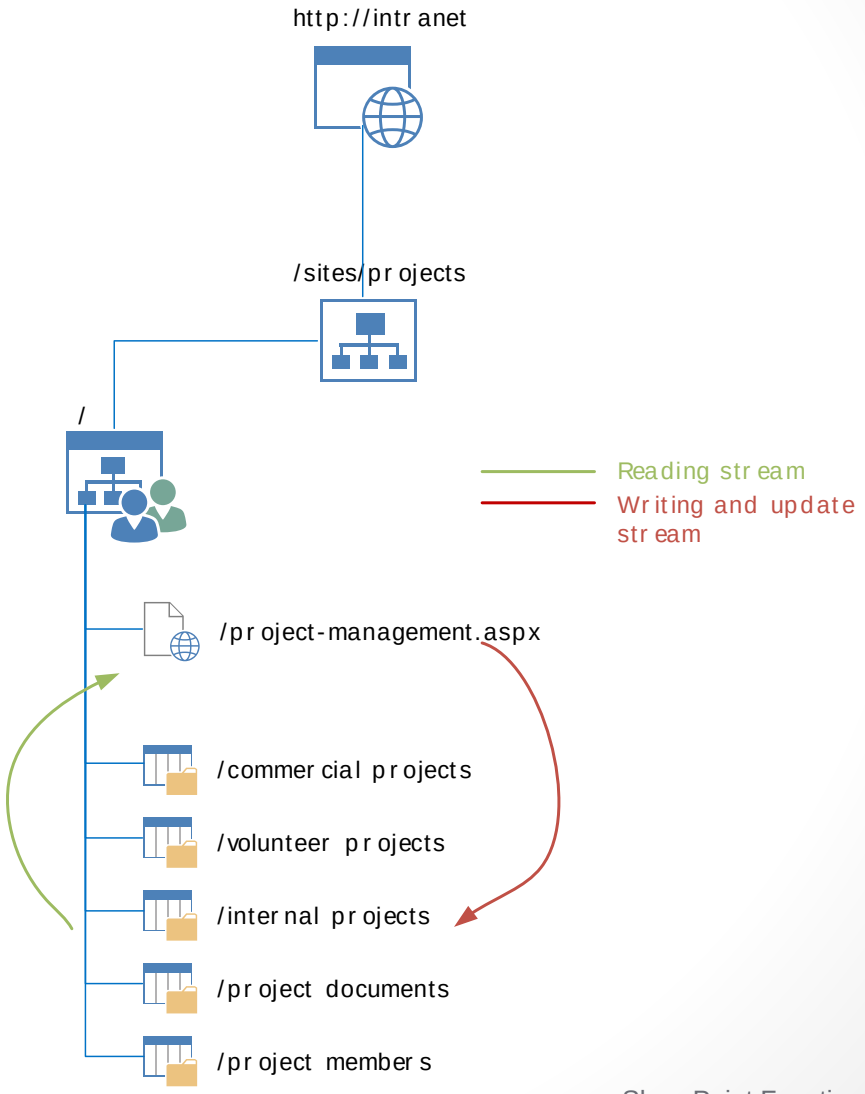
# With SharePoint

3 cases

## Case #1

- Storage in a SharePoint site.
- Access from the same site.

| | ↑ | ↓ |
|---|---|---|
| List View Web Part | ✓ | ✓ |
| Content Query Web Part | ✓ | ✗ |
| Search Core Results Web Part | ✓ | ✗ |
| RSS Viewer | ✓ | ✗ |
| Content Search Web Part | ✓ | ✗ |



5

## Determining the data flow directions

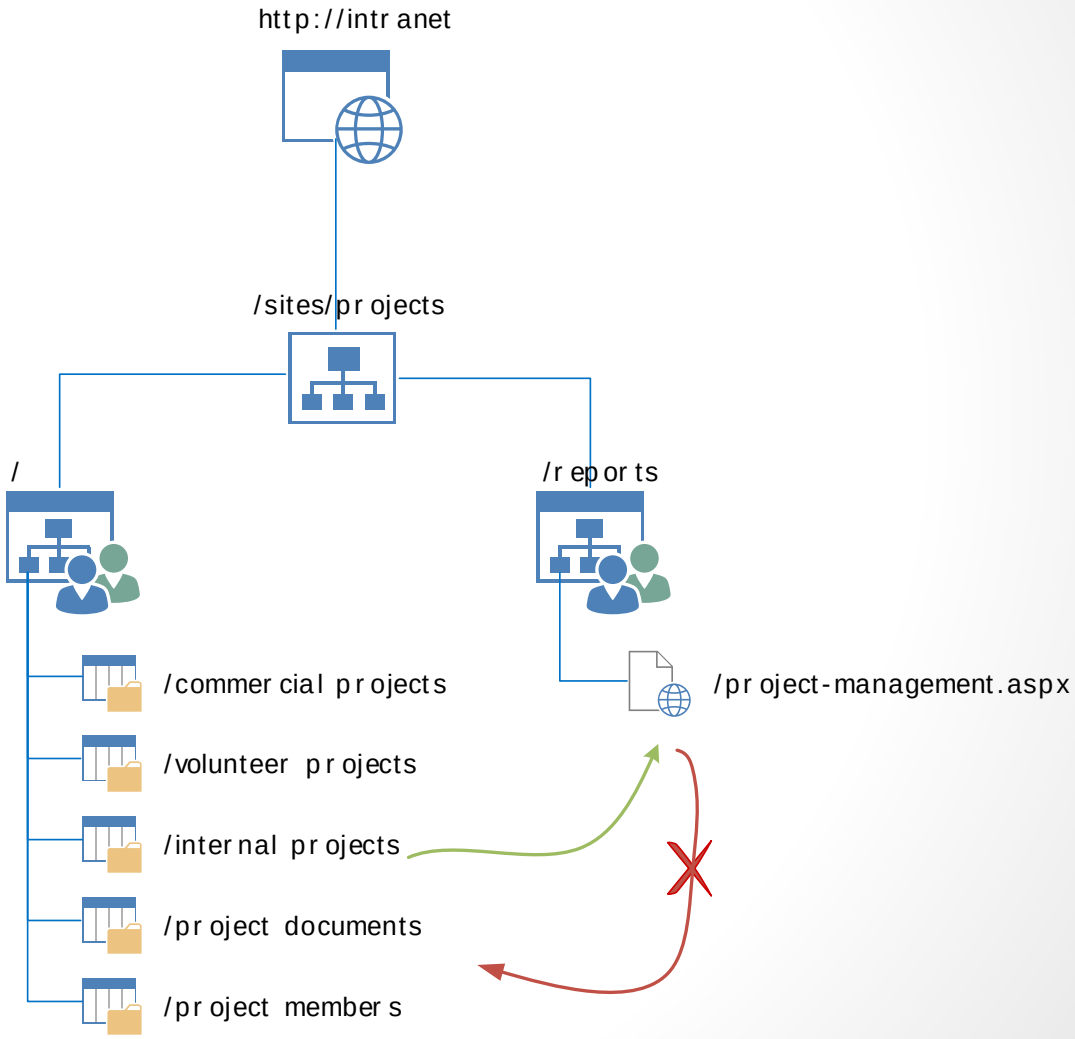
# With SharePoint

3 cases

### Case #2

- Storage in a SharePoint site.
- Access from another site in the same site collection.

| | ↑ | ↓ |
|---|---|---|
| List View Web Part | ✗ | ✗ |
| Content Query Web Part | ✓ | ✗ |
| Search Core Results Web Part | ✓ | ✗ |
| RSS Viewer | ✓ | ✗ |
| Content Search Web Part | ✓ | ✗ |



5

# Determining the data flow directions

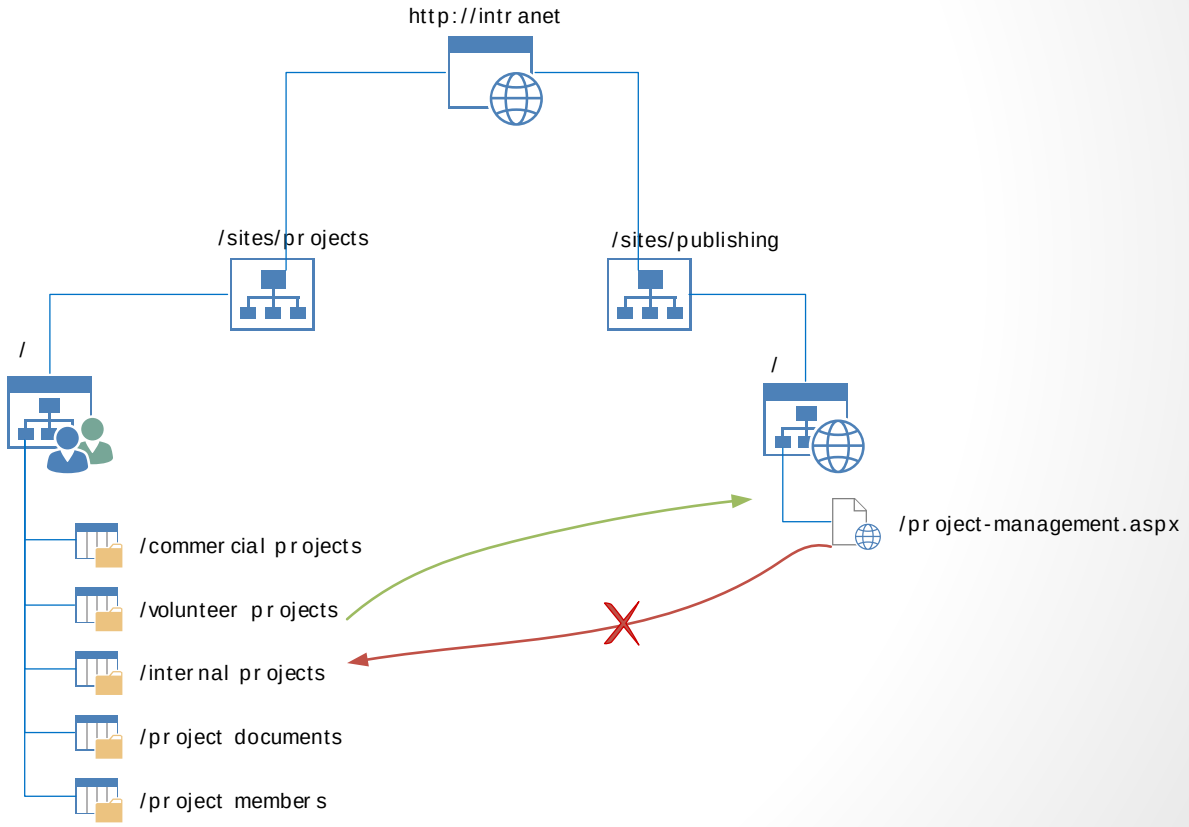
# With SharePoint

## 3 cases

## Case #3

- Storage in a SharePoint site.
- Access from another site in a different site collection.

| | ↑ | ↓ |
|---|---|---|
| List View Web Part | ✗ | ✗ |
| Content Query Web Part | ✗ | ✗ |
| Search Core Results Web Part | ✓ | ✗ |
| RSS Viewer | ✗ | ✗ |
| Content Search Web Part | ✓ | ✗ |



5

## Determining the data flow directions

# With SharePoint

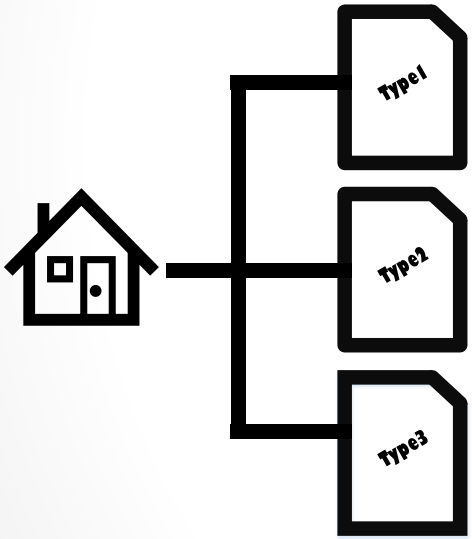
## Components

| | SharePoint 2007 | SharePoint 2010 | SharePoint 2013 | Maximum range | Language used |
|---|---|---|---|---|---|
| Reading flow | ListView Web Part | | | Lists & Libraries | CAML |
| | Content Query Web Part | | | Site collection | CAML |
| | Search Core Results Web Part | | | Web Application | KQL |
| | RSS Viewer | | | Site collection | - |
| | | | Content Search Web Part | Web Application | KQL |
| Writing and updating flow | ListView Web Part | | | Lists & Libraries | CAML |

6

## Define data access point

# According to the data storage distribution and flow, where are the data access points and what types are exposed?



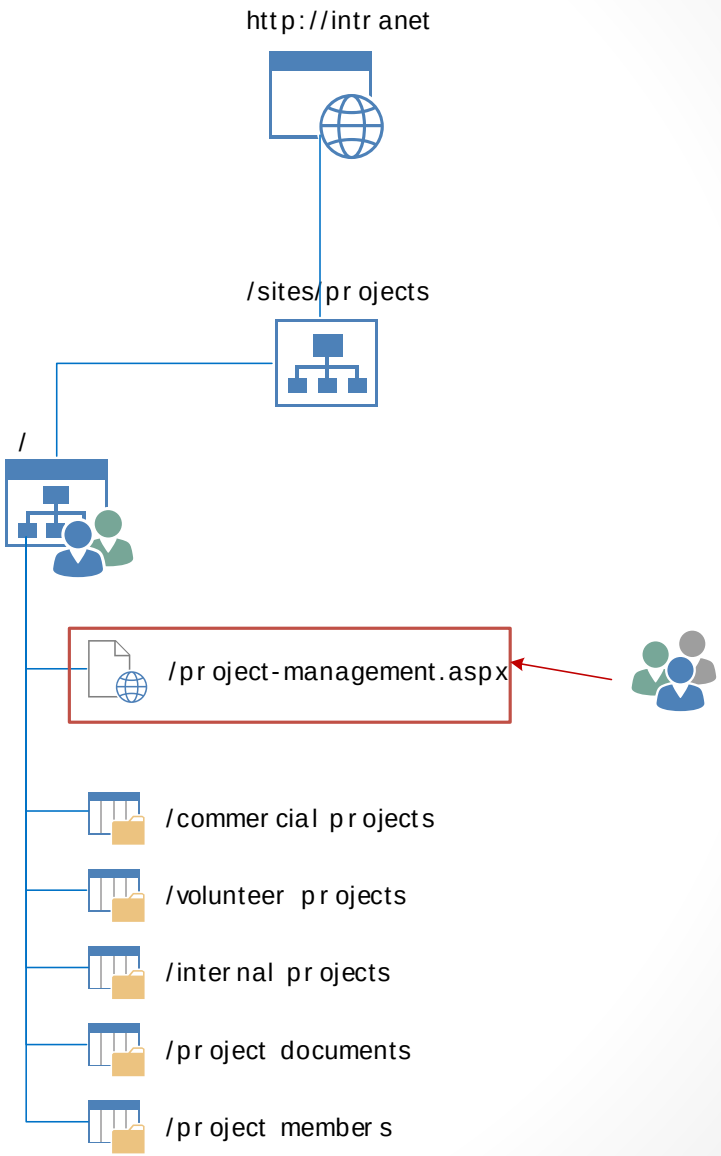

- Raw data
  - Plain text not related to any specific type
    - Ex: A welcome text
- Data aggregation
  - Compilation of data from different data sources
    - Ex: Dashboards
- Typed information
  - Single Item metadata view
  - Visualization of data corresponding to a specific type
    - Ex: A project sheet

6

## Define data access point

# With SharePoint

- Raw data
  - Wiki Pages
- Data aggregation
  - Web Parts Pages
- Typed information
  - List Forms
  - Publishing Pages



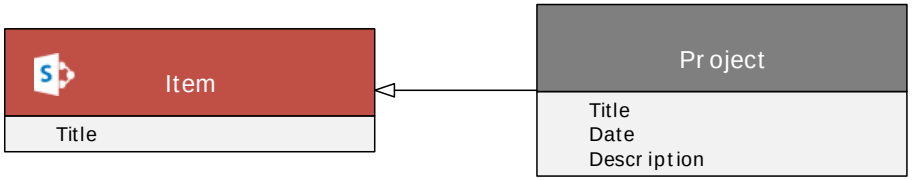
6

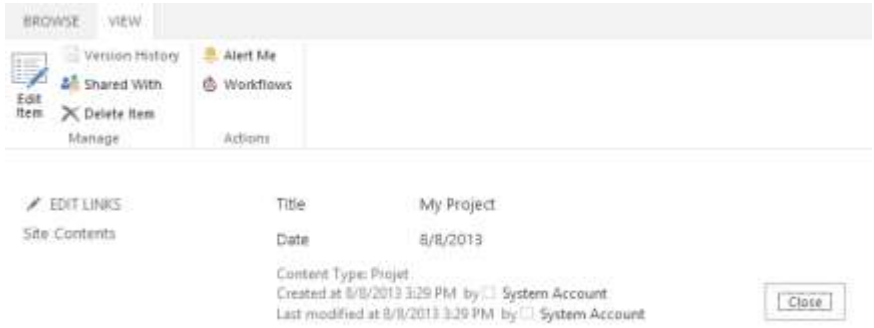
## Define data access point

# With SharePoint

## Typed information

## List Forms



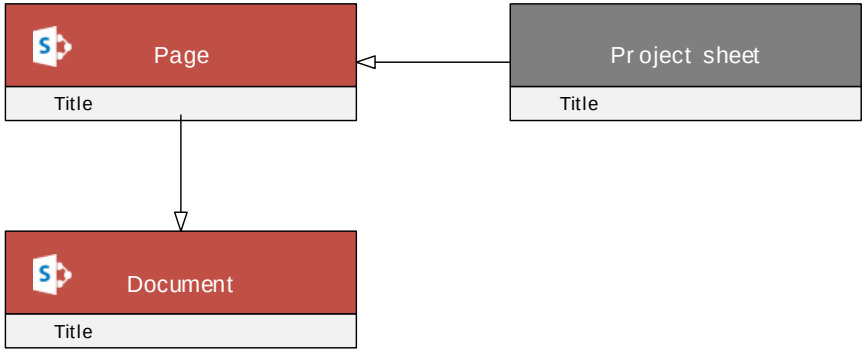
6

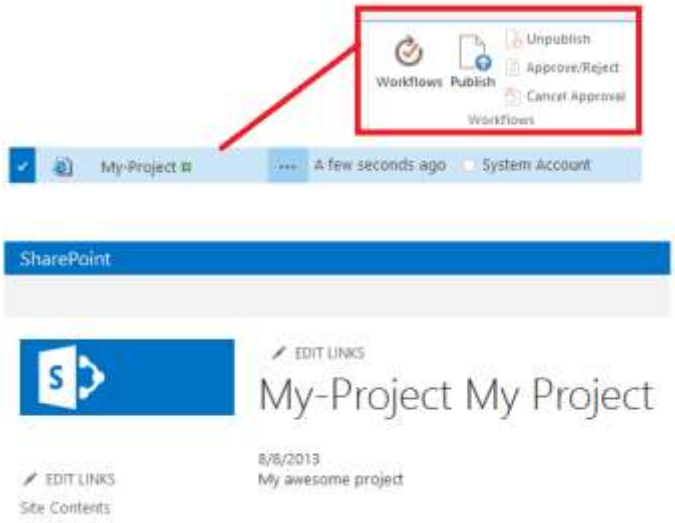
## Define data access point

# With SharePoint

## Typed information

## Publishing page



7

## Define the conditions for information recovery

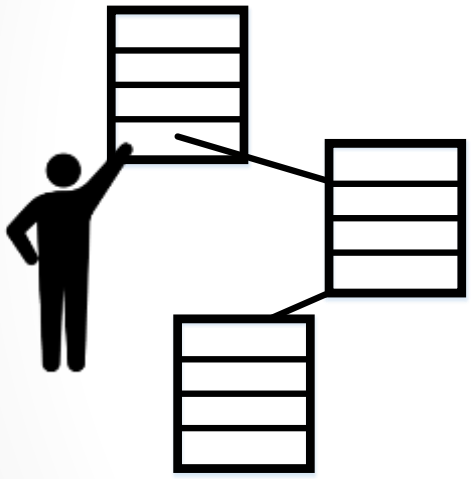
# Under which conditions should information should appear in the page?

- Are the recovered items have to correspond to reality at time T in SharePoint?
- Is there a sort needed on the elements?
- Are there any hierarchical constraints between elements?
- Are there any relationships between entities?
- Are these items must be targeted to a specific audience?
- Are there multilingual constraints?
- Must queries be reused?
- Are there conditional constraints on queries?
- What is the Information life cycle?

8

## Define relational behaviors

# What are the relational constraints on the information?

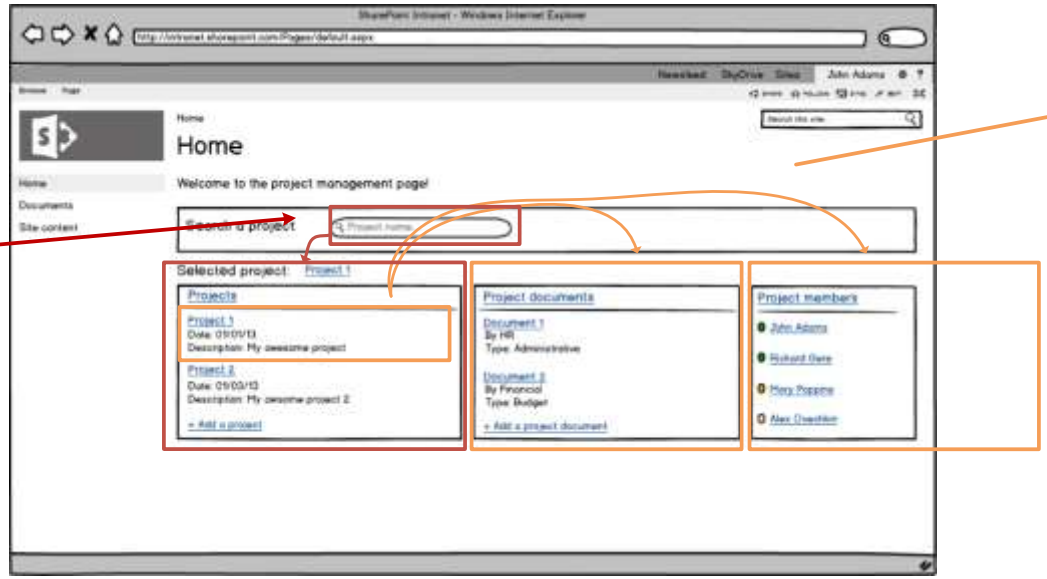


- Notions of filtering and connections in response to actions
  - Ex: An user select an item in a list → an other value is selected in a other component.
  - Ex: Language on the page has changed

8

## Define relational behaviors

# What are the relational constraints on the information?



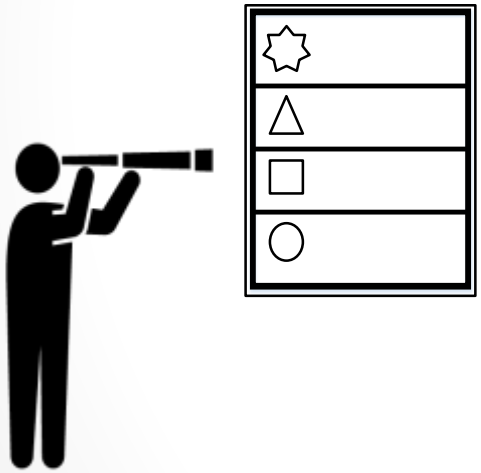

Project filter

Members and documents by project

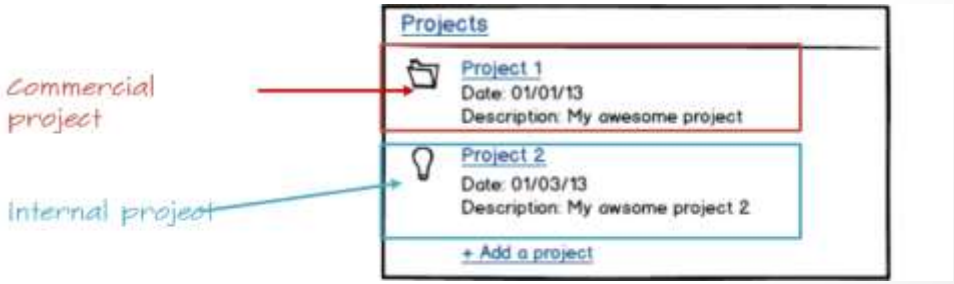
9

## Define the information display

# How is information displayed?



- What are the display details
  - Metadata
  - Style
    - Colours, fonts, etc…
- Should they appear grouped or alone
  - Mixed types inside one container with visual distinction

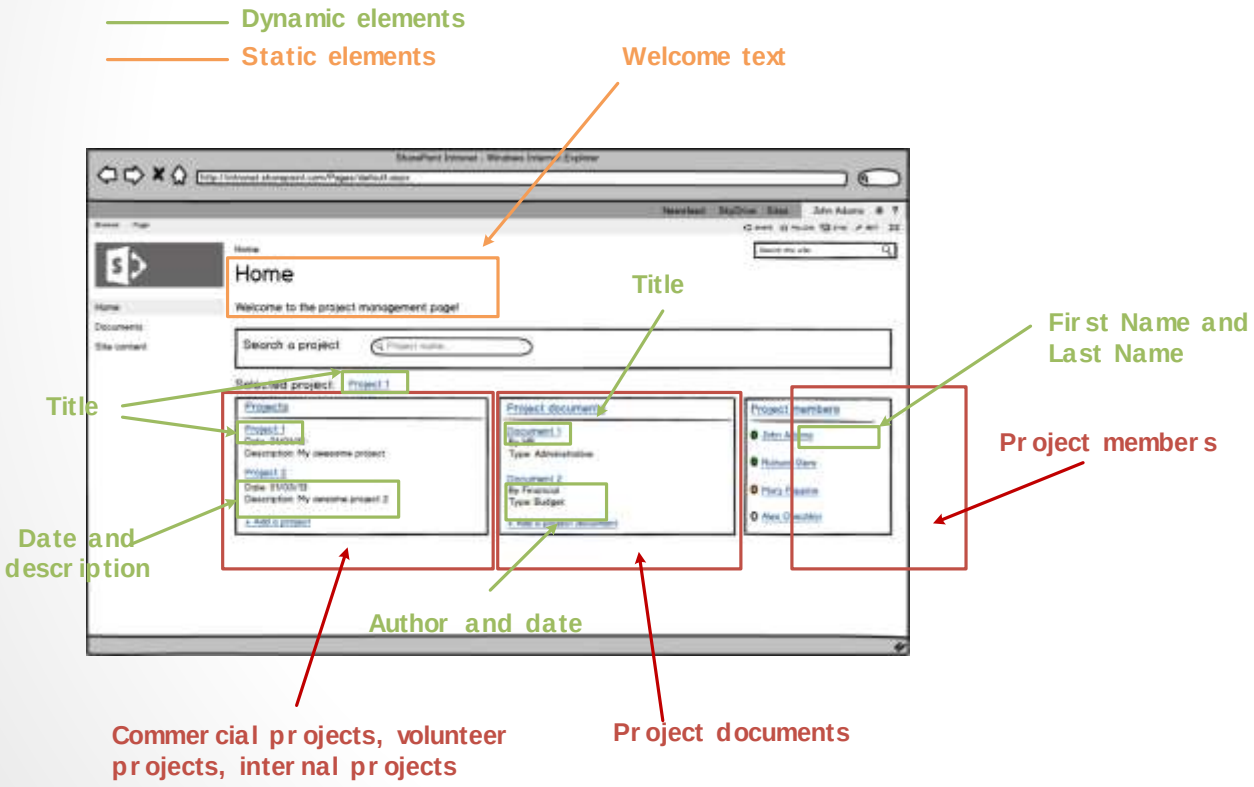


9

# Define the information display

# With SharePoint

## Display management




## XSL

- Limitations due to compiler version
- Standard before SharePoint 2013 (but still present)

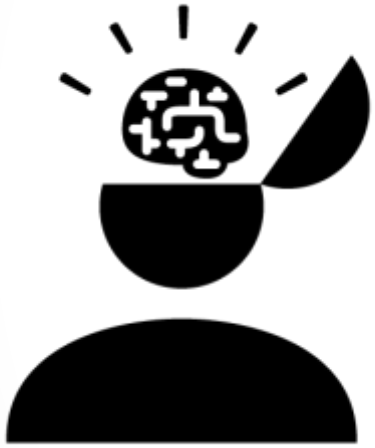
## JavaScript/HTML

- Display templates with result sources (search)
- More powerfull but more complex
- Standard with 2013

B

## Methodology summary

# Your functional analysis with SharePoint

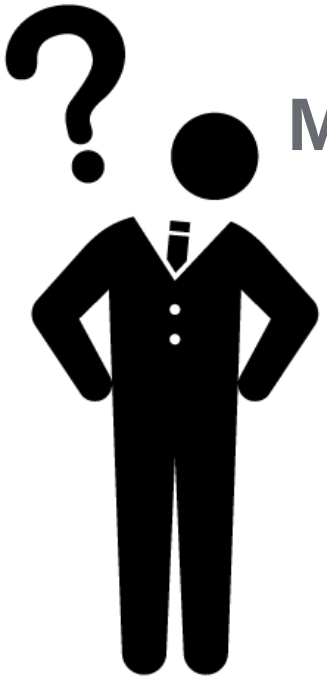


1. What does the requirement look like?
2. What are the information types?
3. What are the relationships?
4. Where data is stored?
5. What are the data flows?
6. What are the page types?
7. What are the display constraints?
8. What are the behaviors on pages?
9. How information are displayed?

# Tools


| If you<br>**Want to** | We<br>**Recommend to you** |
|---|---|
| Draw pretty schemas | Microsoft Visio |
| Define your backlog as a map | Speclog http://www.speclog.net/ |
| Define your backlog as a list | Excel, TFS |
| Draw wireframes | Balsamiq http://balsamiq.com/ |
| Brainstorm about taxonomy | Xmind http://www.xmind.net/ |
| Get more details about this method | GSoft website www.gsoft.com (detailed ebook coming soon!) |

# Conclusion



Merci!

Templates & Examples on SkyDrive

SHAREPOINT SATURDAY OTTAWA

Remember to fill out your evaluation forms to win some great prizes!

&

Join us for SharePint today!

| | |
|---|---|
| Date & Time: | Nov 23rd, 2013 @6:00 pm |
| Location: | The Observatory Pub,<br>Algonquin Student's Association |
| Address: | A-170 on Algonquin Campus |
| Parking: | No need to move your car!* |
| Site: | http://www.algonquinsa.com/ob.aspx |

*Please drive responsibly! We are happy to call you a cab ☺