

# Web Engineering

---

## Web Technologies II + Project Management for Web Applications



# Where are we?

#	Date	Title
1	5 <sup>th</sup> March	Web Engineering Introduction and Overview
2	12 <sup>th</sup> March	Requirements Engineering for Web Applications
3	19 <sup>th</sup> March	Web Application Modeling
4	26 <sup>th</sup> March	Web Application Architectures
5	16 <sup>th</sup> April	Developing Applications with WebML
6	23 <sup>rd</sup> April	Testing and Usability
7	30 <sup>th</sup> April	Web Technologies I
8	7 <sup>th</sup> May	Web Application Security
9	21 <sup>th</sup> May	Web Application Development Process
<b>10</b>	<b>28<sup>th</sup> May</b>	<b>Web Technologies II</b>
<b>11</b>	<b>11<sup>th</sup> June</b>	<b>Project Management for Web Applications</b>
12	18 <sup>th</sup> June	Mobile Application Development
13	25 <sup>th</sup> June	Final Exam



- Web Technologies
  - JQuery
  - AngularJS
- Project Management for Web Applications
  - Introduction
  - Project Management For Web Applications
    - Challenges
    - Managing the Development Team
    - Managing the Development Process
    - Project Risks & Risk Management
  - Wrap-up



# JQUERY

*Acknowledgements*

<http://www.w3schools.com/jquery/>

[www.webstepbook.com/supplements-2ed/slides/ppt/22-jQuery1.pptx](http://www.webstepbook.com/supplements-2ed/slides/ppt/22-jQuery1.pptx)

- jQuery is a lightweight, "write less, do more", JavaScript library.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.
- jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
- jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

- Write less, do more:
  - `$(".p.neat").addClass("ohmy").show("slow");`
- Performance
- Plugins
- It's standard
- ... and fun!
  
- Many of the biggest companies on the Web use jQuery, such as:
  - Google
  - Microsoft
  - IBM
  - Netflix

```
$(document).ready(function(){  
    $("p").click(function(){  
        $(this).hide();  
    });  
});
```

<p>If you click on me, I will disappear.</p>

<p>Click me away!</p>

<p>Click me too!</p>

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event methods
- Effects and animations
- AJAX
- Utilities



- **Identification:** how do I obtain a reference to the node that I want.
- **Traversal:** how do I move around the DOM tree.
- **Node Manipulation:** how do I get or set aspects of a DOM node.
- **Tree Manipulation:** how do I change the structure of the page.

Two ways to start using jQuery on your web site:

- Download the jQuery library from [jQuery.com](http://jquery.com)

```
<head>  
<script src="jquery-1.11.3.min.js"></script>  
</head>
```

- Include jQuery from a content delivery network, like Google

```
<head>  
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.3/jquery.min.js"></script>  
</head>
```

The jQuery syntax is tailor made for **selecting** HTML elements and performing some **action** on the element(s).

Basic syntax is:  **$\$(selector).action()$**

- A \$ sign to define/access jQuery
- A (*selector*) to "query (or find)" HTML elements
- A jQuery *action()* to be performed on the element(s)

- jQuery selectors allow you to select and manipulate HTML element(s).
- jQuery selectors are used to "find" (or select) HTML elements based on their id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.
- All selectors in jQuery start with the dollar sign and parentheses: \$().
- \$("p") – select all <p> elements on a page

- An event represents the precise moment when something happens (moving a mouse over an element, selecting a radio button, etc.)
- E.g.: On click to a paragraph do something

```
$("#p").click(function(){  
    // action goes here!!  
});
```



- jQuery Hide/Show
- jQuery Fade
- jQuery Slide
- jQuery Animate
- jQuery stop()
- jQuery Callback
- jQuery Chaining

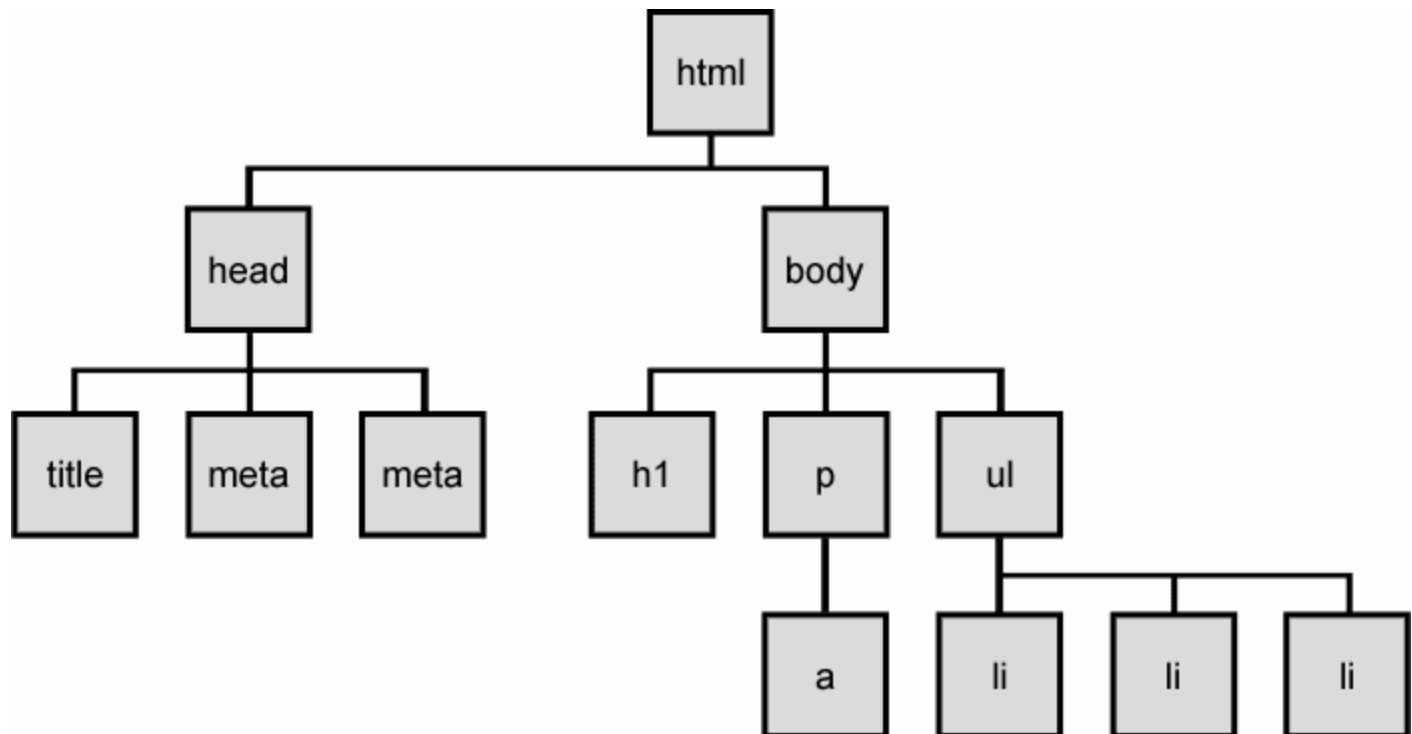
- One very important part of jQuery is the possibility to manipulate the DOM.
- jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.
  
- jQuery Get
- jQuery Set
- jQuery Add
- jQuery Remove
- jQuery CSS Classes
- jQuery css()
- jQuery Dimensions

name	description
<a href="#">getElementById</a>	returns array of descendents with the given tag, such as "div"
<a href="#">getElementsByTagName</a>	returns array of descendents with the given tag, such as "div"
<a href="#">getElementsByName</a>	returns array of descendents with the given name attribute (mostly useful for accessing form controls)
<a href="#">querySelector</a> *	returns the first element that would be matched by the given CSS selector string
<a href="#">querySelectorAll</a> *	returns an array of all elements that would be matched by the given CSS selector string



DOM method	jQuery equivalent
<code>getElementById("id")</code>	<code>\$("#id")</code>
<code>getElementsByTagName("tag")</code>	<code>\$("tag")</code>
<code>getElementsByName("somename")</code>	<code>\$("[name='somename']")</code>
<code>querySelector("selector")</code>	<code>\$("selector")</code>
<code>querySelectorAll("selector")</code>	<code>\$("selector")</code>

- jQuery traversing are used to "find" (or select) HTML elements based on their relation to other elements. Start with one selection and move through that selection until you reach the elements you desire.



jQuery methods for traversing the DOM tree are:

- `parent()`
- `parents()`
- `parentsUntil()`
- `children()`
- `find()`
- `siblings()`
- `next()`
- `nextAll()`
- `nextUntil()`
- `prev()`
- `prevAll()`
- `prevUntil()`
- `first()`, `last()` and `eq()`

- jQuery provides several methods for AJAX functionality.
- With the jQuery AJAX methods, you can request text, HTML, XML, or JSON from a remote server using both HTTP Get and HTTP Post - And you can load the external data directly into the selected HTML elements of your web page!
- **jQuery load()**
  - loads data from a server and puts the returned data into the selected element.
  - **Syntax:**
  - `$(selector).load(URL,data,callback);`
- **jQuery \$.get()**
  - The \$.get() method requests data from the server with an HTTP GET request.
  - **Syntax:**
  - `$.get(URL,callback);`
- **jQuery \$.post()**
  - The \$.post() method requests data from the server using an HTTP POST request.
  - **Syntax:**
  - `$.post(URL,data,callback);`



# ANGULARJS

*Acknowledgements*

<http://www.w3schools.com/angular/>

- A JavaScript framework for creating dynamic web applications
- Open Source
  - GitHub: <https://github.com/angular/angular.js>
- AngularJS extends HTML with new attributes.
- AngularJS is perfect for Single Page Applications (SPAs).

- AngularJS version 1.0 was released in 2012.
- Miško Hevery, a Google employee, started to work with AngularJS in 2009.
- AngularJS project is officially supported by Google.



AngularJS can be added to an HTML page with a `<script>` tag.

```
<script  
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.mi  
n.js"></script>
```



AngularJS extends HTML with **ng-directives**.

- The **ng-app** directive defines an AngularJS application.
- The **ng-model** directive binds the value of HTML controls (input, select, textarea) to application data.
- The **ng-bind** directive binds application data to the HTML view.
- The **ng-controller** directive defines the application controller.

- ```
<!DOCTYPE html>
<html>
<script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.3.14/angular.min.js
"></script>
<body>

<div ng-app="">
  <p>Name: <input type="text" ng-model="name"></p>
  <p ng-bind="name"></p>
</div>

</body>
</html>
```

AngularJS filters can be used to transform data:

Filter	Description
currency	Format a number to a currency format.
filter	Select a subset of items from an array.
lowercase	Format a string to lower case.
orderBy	Orders an array by an expression.
uppercase	Format a string to upper case.

- **\$http** is an AngularJS service for reading data from remote servers.
- **ng-repeat** directive is used for displaying tables.
- **ng-disabled** directive binds AngularJS application data to the disabled attribute of HTML elements.
- **ng-click** directive defines an AngularJS click event.

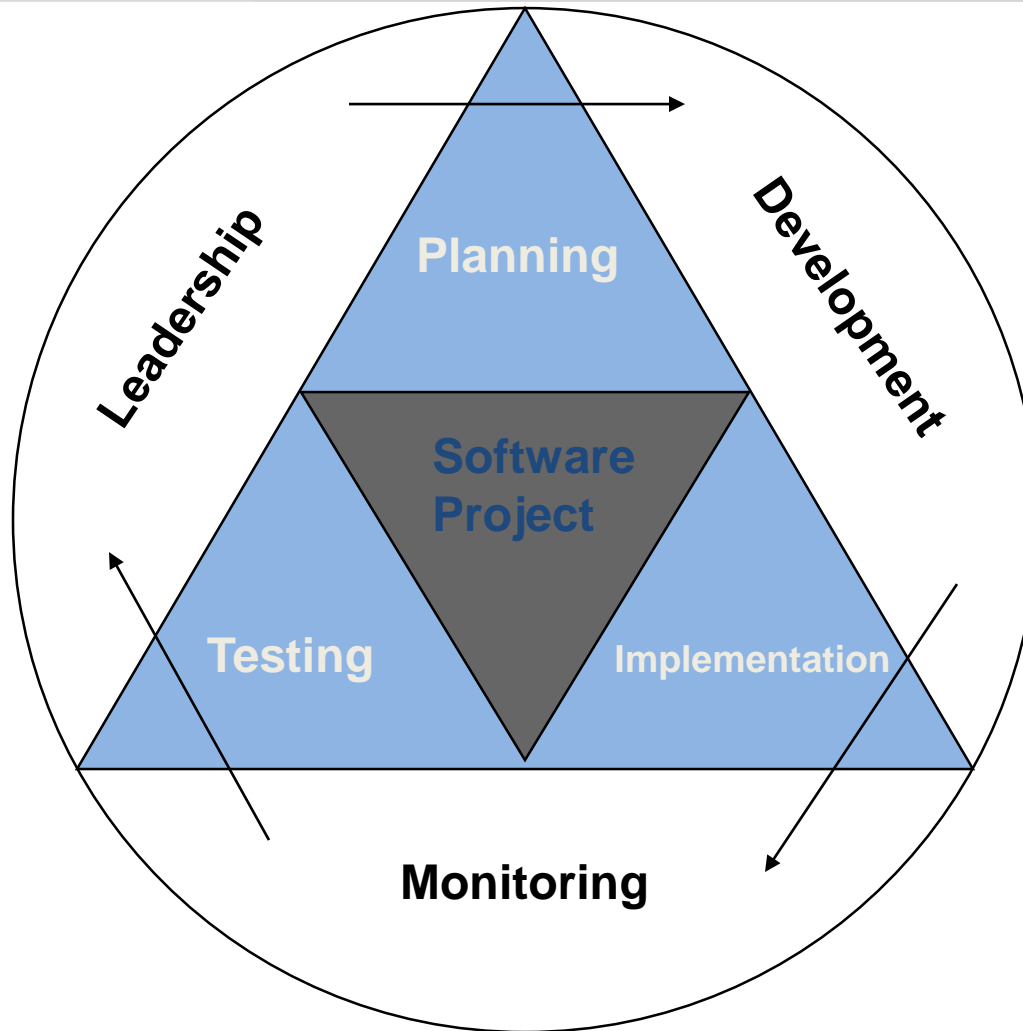
- The AngularJS Global API is a set of global JavaScript functions for performing common tasks like:
  - Comparing objects
  - Iterating objects
  - Converting data
- The Global API functions are accessed using the angular object.
- E.g. `angular.lowercase()` - Converts a string to lowercase



# PROJECT MANAGEMENT

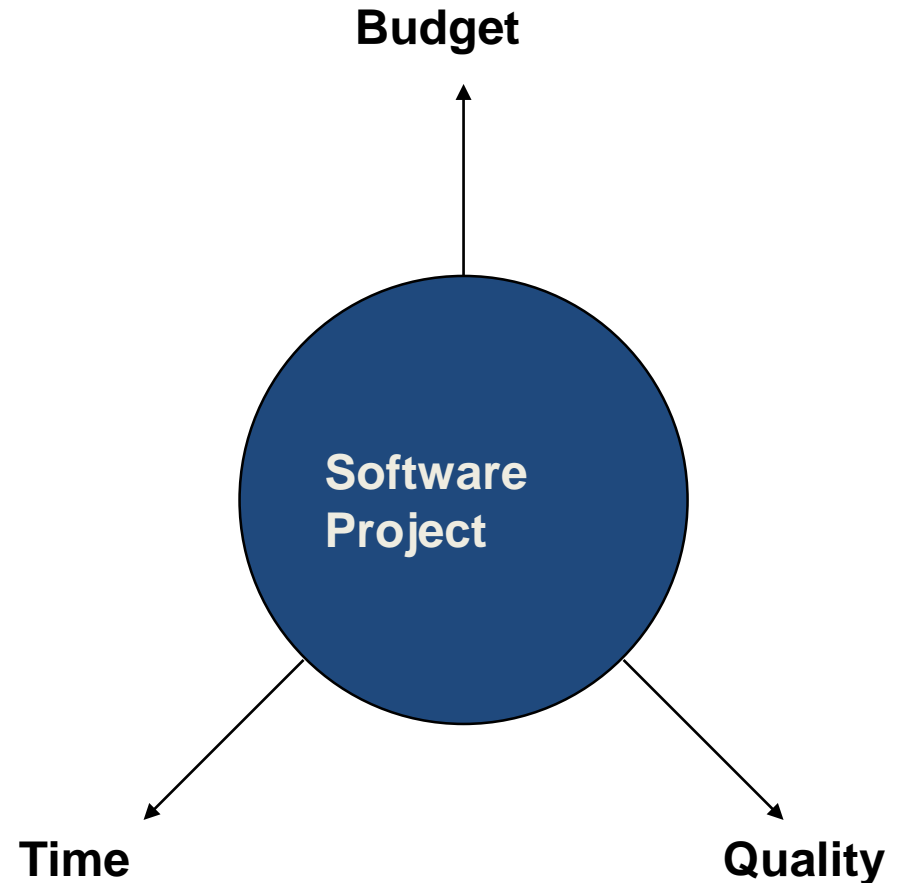
- Project management
  - A systematic approach to planning and guiding project processes from beginning to end.
  - Project management is the discipline of **planning, organizing, securing** and **managing** resources to bring about the successful completion of specific project **goals** and **objectives**<sup>1</sup>.
- It is usually a human-centered activity.
- Like in requirements analysis, conflict resolution is critical
- Many development teams are still “new” to the Web
  - Short history, inexperienced in management
  - Experienced in traditional software only

<sup>1</sup>[http://en.wikipedia.org/wiki/Project\\_management](http://en.wikipedia.org/wiki/Project_management)





- A change in one goal incurs trade-offs with the others.
- Be sure all stakeholders are aware of this relationship!
- Agile methods are introducing additional variable – scope
  - Scope reductions can reduce budget, shorten the development time and rise the quality level.



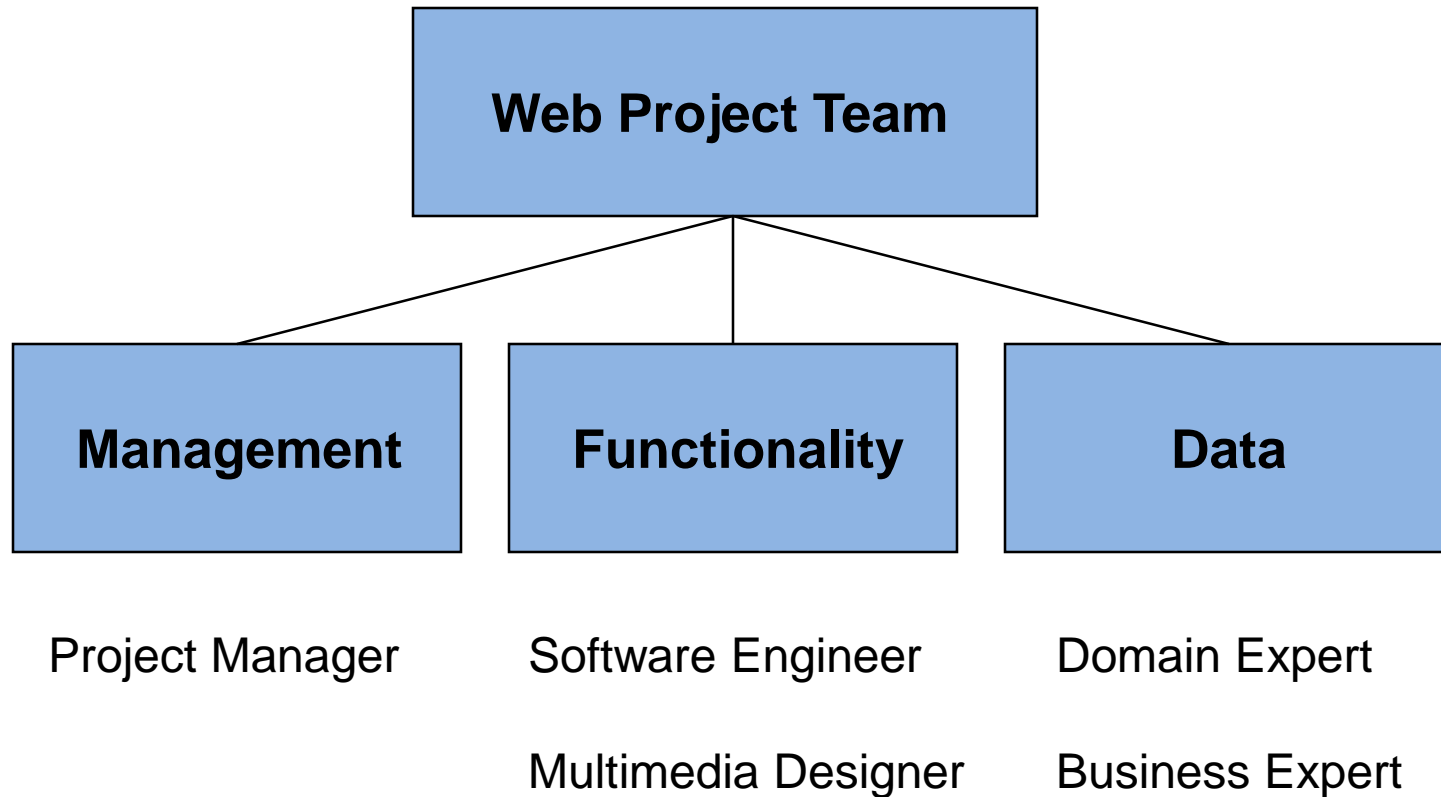
- In Traditional Projects:
  - Quality product at lowest cost
  - 10-100 team members
  - 12-18 month horizon
  - Requirements-based; structured phases; document-driven
  - OO methods
  - Rigid processes
  - Complex; poor reusability
  - Experienced, professional developers
- In Web Projects:
  - Usable product in shortest time
  - 3-10 team members
  - 3-6 month horizon
  - Agile methods; prototypes
  - Component-based methods; multimedia; visual programming
  - Ad-hoc processes
  - Standardized; high reusability
  - Multimedia designers; Web programmers; marketers

- Leadership
  - Poor/incomplete planning
  - Unique/legacy software systems
  - Highly technical leadership
- Development
  - Individuality
  - Many alternative solutions
  - Rapid change
- Monitoring
  - Scope of monitoring concerns
  - Correctness of monitored values

- **Novelty**
  - Unknown & uninformed audiences
- **Usability**
  - No manuals, time to learn extremely short (the app must be intuitive)
- **Dynamics**
  - Time pressures
- **Parallelism**
  - Subteams and inter-communication
- **Continuity**
  - Development to transition
- **Juvenility**
  - Youth; less experience
- **Immaturity**
  - Inadequate tools

- Web apps are “simple”
- Aesthetics
- Spontaneity
- Ubiquity
- Compatibility
- Stability & Security
- Scalability

- Success is largely determined by group dynamics & how well they are managed
  - Communication among team members
  - Motivation & coordination by the Project Manager
  - Identification & resolution of conflicts ASAP
  - Concurrent engineering
  - Primary & backup strategies
  - Documentation is everyone's responsibility



- Mediator
- Motivator
- Communicator
- Translator
- Trainer
- Customer liaison
  - Requirements during development
  - Post-deployment?



1. Promote the professional self-conception of each team member and ethical behavior.
2. Stress the importance of different app knowledge
3. Solve conflicts quickly
4. Keep roles well-defined
5. Look for parallel developments & synergies
6. Spread the documentation task fairly
7. Promote & coordinate continuous use of tools
8. Translate costs & values
9. Keep the customer involved
10. Monitor project progress and objectives

- Tool-driven vs. document-driven
  - Requirements & test suites
  - Communication (Slack, Wikis, etc.)
- Configuration management
  - Versioning
  - Short iteration cycles
  - No project is too small for it!
- Measuring progress
  - System specification
  - The application itself

- Risk: the probability of an activity to result in a loss
- Most critical risks:
  - Personnel deficits
  - Unrealistic time and cost specs
  - Incompatible third-party components
  - Misunderstood properties
  - Poor user interface

- Unclear definition of objectives
- Wrong target audience
- Development-oriented page structure
- Inconsistent design
- Insufficient budget for maintenance
- Content recycling & poor linking
- Mixing Internet & intranet
- Marketing research is seen as usability research


- How likely is a problem to occur, what will be the impact, and what are the solutions?
- Assessment: Identify, analyze, & prioritize
- Control: Provision, monitor, mitigate
- Groups are better at assessing and managing risk than individuals
- Perform a cost-benefit analysis to justify risk management activities

# WRAP-UP

- jQuery a lightweight, "write less, do more", JavaScript library
- AngularJS a JavaScript framework for creating dynamic web applications
- Project Management is part of the meta-development process (process about the process)
  - Minimize risks
  - Enable development process monitoring
  - Require integration with the development process ("probe" points)

- Mandatory reading
  - Kappel, G., Proll, B. Reich, S. & Retschitzegger, W. (2006). *Web Engineering*, Wiley & Sons. **9<sup>th</sup> Chapter**



#	Date	Title	
1	5 <sup>th</sup> March	Web Engineering Introduction and Overview	
2	12 <sup>th</sup> March	Requirements Engineering for Web Applications	
3	19 <sup>th</sup> March	Web Application Modeling	
4	26 <sup>th</sup> March	Web Application Architectures	
5	16 <sup>th</sup> April	Developing Applications with WebML	
6	23 <sup>rd</sup> April	Testing and Usability	
7	30 <sup>th</sup> April	Web Technologies I	
8	7 <sup>th</sup> May	Web Application Security	
9	21 <sup>th</sup> May	Web Application Development Process	
10	28 <sup>th</sup> May	Web Technologies II	
11	11 <sup>th</sup> June	Project Management for Web Applications	
	<b>12</b>	<b>18<sup>th</sup> June</b>	<b>Mobile Application Development</b>
	13	25 <sup>th</sup> June	Final Exam

# Questions?

