

<https://www.halvorsen.blog>



Week Assignment

Software Platforms

Hans-Petter Halvorsen

Week Assignment

1. **Create HTML Web Site**
2. **Sprint Review** of previous Iteration (Alpha Release)
3. **Sprint Planning** of Next Iteration (Beta Release)
4. **Implementing** (If you haven't started the coding, now its the time!!!) The next 4 weeks our main focus will be Implementation/Coding of our System

We will use **Azure DevOps** as our Project Tool.

Here we will maintain the Product Backlog and the Sprint Backlog, use the Taskboard, etc.



Software Platforms

The Digital Age

Smartphone, 2007

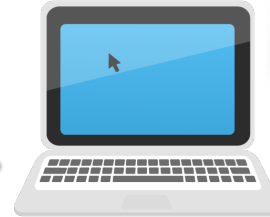


1984: Macintosh

1976: Apple I



World Wide Web,
1989-93



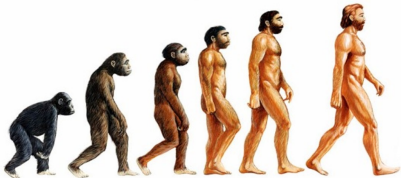
PC, 1981
(IBM)

The
Microprocessor,
1971

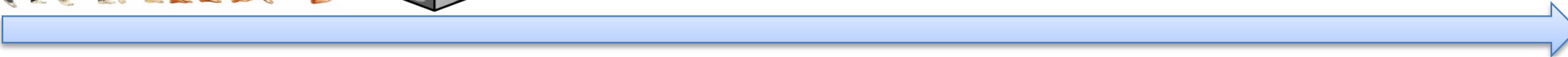


Internet,
1968-91

The first Computer
The Turing
machine, 1936



Internet of
Things (IoT)
and Robots
(Artificial
Intelligence
and Machine
Learning)
take over the
World?



Software Platforms

Server Platforms



Database Platforms

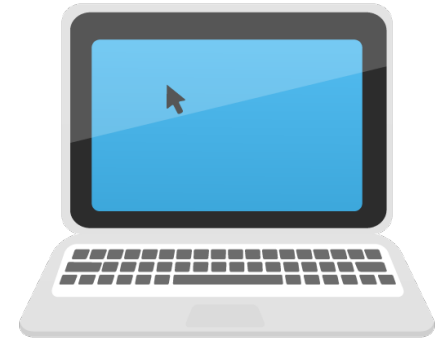
Web Server Platforms

Cloud Platforms



Web Platforms

Mobile
Platforms



Desktop Platforms

Desktop Platforms?

(Operating Systems)

Desktop Platforms

(Operating Systems)



Windows



macOS



Linux



Chrome OS

Mobile Platforms?

(Smartphone/Tablet Operating Systems)

Mobile Platforms

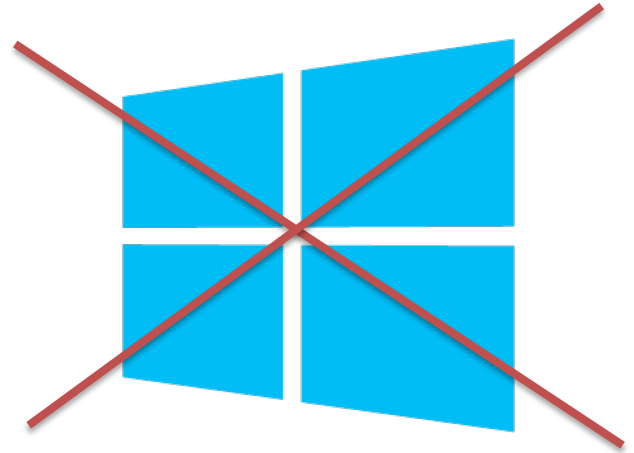
(Smartphone/Tablet Operating Systems)



Android



iOS



Windows 10 Mobile

Web Browsers?

Web Browsers



Internet Explorer



Microsoft Edge



Safari



Chrome



Firefox



Opera

Web Browser - The 3 big ones



Chrome

Windows + macOS + iOS



Edge

Windows + macOS + iOS



Safari

macOS + iOS

1.4+ billion active iOS devices



Den store feil fremtidsforskere og datafreakere gjør, er å trekke erfaringene fra bruk av PC på arbeidsplassene inne i hjemmene. En slik projisering holder ikke, skriver *Leif Osvoold* i Oslo.

DEBATT

Internett en flopp!

Dataekspertene og såkalte fremtidsforskere spør i økende grad at Internett vil bli dominerende i vårt dagligliv i de nærmeste årene, fordi vi vil bli nødt til å ta den i bruk via vår hjemme-PC. Tilbakvisning av slike gale trend-baserte påstander er nå nødvendige, og her er en mot-hypotese: Internett er en flop; det vil si en «motegreie» som kommer til å dø ut om et par år.

Det er tre grunner til dette: 1) ingen av aktørene på nettet vil tjene penger på å legge seg der med sine tilbud, 2) privat bruk av nettet vil være marginalt, og 3) mengden av informasjon på nettet vil bli så enorm at det vil skape frustrerende store søkeproblemer, og dermed frafall av brukere. Hva gjelder punkt 2 så tror jeg at vi snart vil få se en leverandørflukt fra Internett, når disse oppdager at de har lurt seg selv, redde for ikke å være moderne eller være tilstede der «alle de andre» er.

Hva gjelder punkt 2 så vil jeg ta utgangspunkt i hva flere medieguruer sier. De uttaler at Internett innen år 2000 vil være en like naturlig del av dagliglivet som PC-en er idag. Dette postulatet er galt, av den enkle grunn at PC idag ikke er en naturlig del av dagliglivet. Feilen oppstår fordi man ikke skiller mellom bruk av PC på jobb, og hjemme.

Datautviklingen og bruk av PC på jobb har revolusjonert arbeidslivet, og vil fortsatt gjøre det. Dette er stort sett en velsig-



Motegreie. Internett er en motegreie som kommer til å dø ut om et par år, mener *Leif Osvoold*.

nelse for arbeidstagerne. Den store feil fremtidsforskerne og datafreakene gjør er imidlertid å trekke erfaringene fra bruk av PC på arbeidsplassene inn i hjemmet. De påstår derfor at en tilsvarende revolusjon vil skje der, men en slik projisering holder ikke, hverken i teori eller praksis. Utbredelsen av PCer i hjemmene vil nok øke noe i årene som kommer, men jeg tror vi allerede er nær toppen. Og denne toppen er kanskje fem prosent av befolkningen, mens med andre ord 95 prosent ikke bruker PC hjemme (selv om mange har kjøpt en). Og

det er bruken av PC som teller, ikke besittelsen. Grunnen til dette er simpelthen at mennesket er et sosialt vesen, og etter en stund kommer til å bli lei av å kommunisere med en maskin i fritiden. PC i hjemmet kommer i all hovedsak til å bli benyttet til jobb- og studie-relaterte oppgaver, samt til spill og underholdning. Og selv volumet av disse positive anvendelsene blir små, også på lang sikt.

Det er forbløffende å konstatere hvordan fremtidsforskere og dataekspertene overser dette fundamentale sosiale element hos mennesket. Det er enkelt å registrere at vi mennesker er slik skapt at vi faktisk ikke ønsker å forholde oss til en datamaskin hele dagen, men at vi trenger å kommunisere med andre levende vesener. Særlig gjelder det dem som bruker PC på jobben. Vi vil derfor ikke benytte en maskin når vi i fritiden skal kommunisere med omverdenen. Vi vil heller ikke sitte alene hjemme og utføre jobben vår, uten kommunikasjon med et kollegialt arbeidsmiljø. Såkalt «fernarbeid» kommer derfor heller aldri til å bli særlig utbredt, men forbli en ubetydelig avart.

Mennesket vil alltid knytte seg til et sosialt felles miljø, fordi det hører til de basale holdninger og behov vi er skapt med. Disse vil ikke forandre seg i nevneverdig grad, til tross for PCen. Når vi skal bestille våre billetter eller reiser så vil vi snakke med et le-

vende menneske, ikke taste inn på en maskin. Når vi leier vår video så vil vi besøke utleieren og velge i visuelle omgivelser. Shopping vil vi gjøre ved å oppsøke det levende miljø i butikkene, ikke sitte hjemme og bestille varer. Vi klarer ikke å «snakke» med eksterne familiemedlemmer eller venner via en PC, så lenge vi kan ringe eller besøke dem. Vi vil ikke lese hverken aviser, fag- eller skjønnlitteratur ved å «bla» i en datamaskin, men ved å kjenne papiret og boken i våre egne hender. Disse tingene vil ikke kunne erstattes av «PC-opplevelser», og slik vil det heldigvis fortsette å være, for slik er den menneskelige natur. Kort oppsummert: de sosiale basis-behov hos oss står i direkte motstrid til bruk av datasystemer i hjemmet, og vil naturligvis seire i det lange løp. Og når det gjelder bruk av Internett for å få all verdens informasjon, så tror jeg at dette vil dø ut av seg selv. Vi er allerede overført med informasjon, og får dessuten den vi trenger via trykte medier, radio og TV.

Idag er det kun én prosent av befolkningen som bruker Internett hjemme, og særlig flere tror jeg ikke det vil bli. Hvordan det er mulig å lage så mye styr omkring et medium som 99 prosent av folket ikke benytter seg av privat, kan bare forklares med at det er massemediene som er hovedaktørene også på Internett.

1996

”Det var i alle fall det som var spådommen i et leserinnlegg som stod på trykk i Dagens Næringsliv i 1996. I dag bruker vi internett til det meste - og nettbruken fortsetter å øke med 40 prosent i året.”

”Forfatteren av leserinnlegget innrømmet ti år senere at han hadde bommet på spådommen”

<http://www.dn.no/nyheter/2005/10/25/-internett-er-en-flopp>

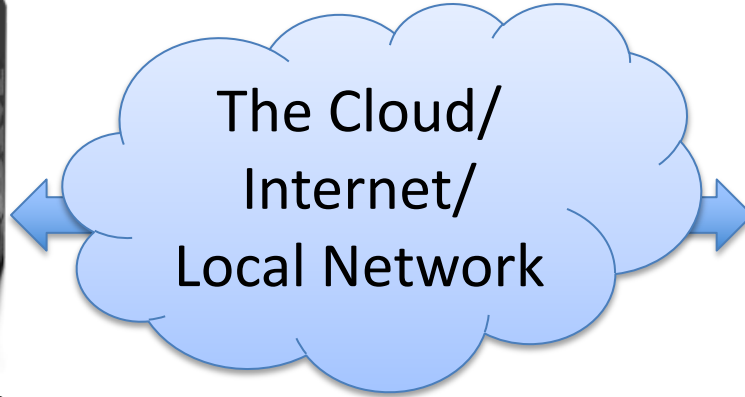
Web Platforms

Servers



Web Servers

Database Servers



Clients



PCs with Web Browsers

Web Server Platforms?

Web Server Platforms



(pronounced "engine x")
- Has become very popular lately

We will use Apache this Week when creating HTML Web Site



Apache

PHP

Cross-platform: UNIX, Linux, OS X, Windows, ...

The term web server can refer to either the hardware (the computer) or the software (the computer application) that helps to deliver web content that can be accessed through the Internet.

The most common use of web servers is to host websites, but there are other uses such as gaming, data storage or running enterprise applications.

We will use IIS later when creating a Test Environment



Microsoft IIS

Internet Information Services

ASP.NET

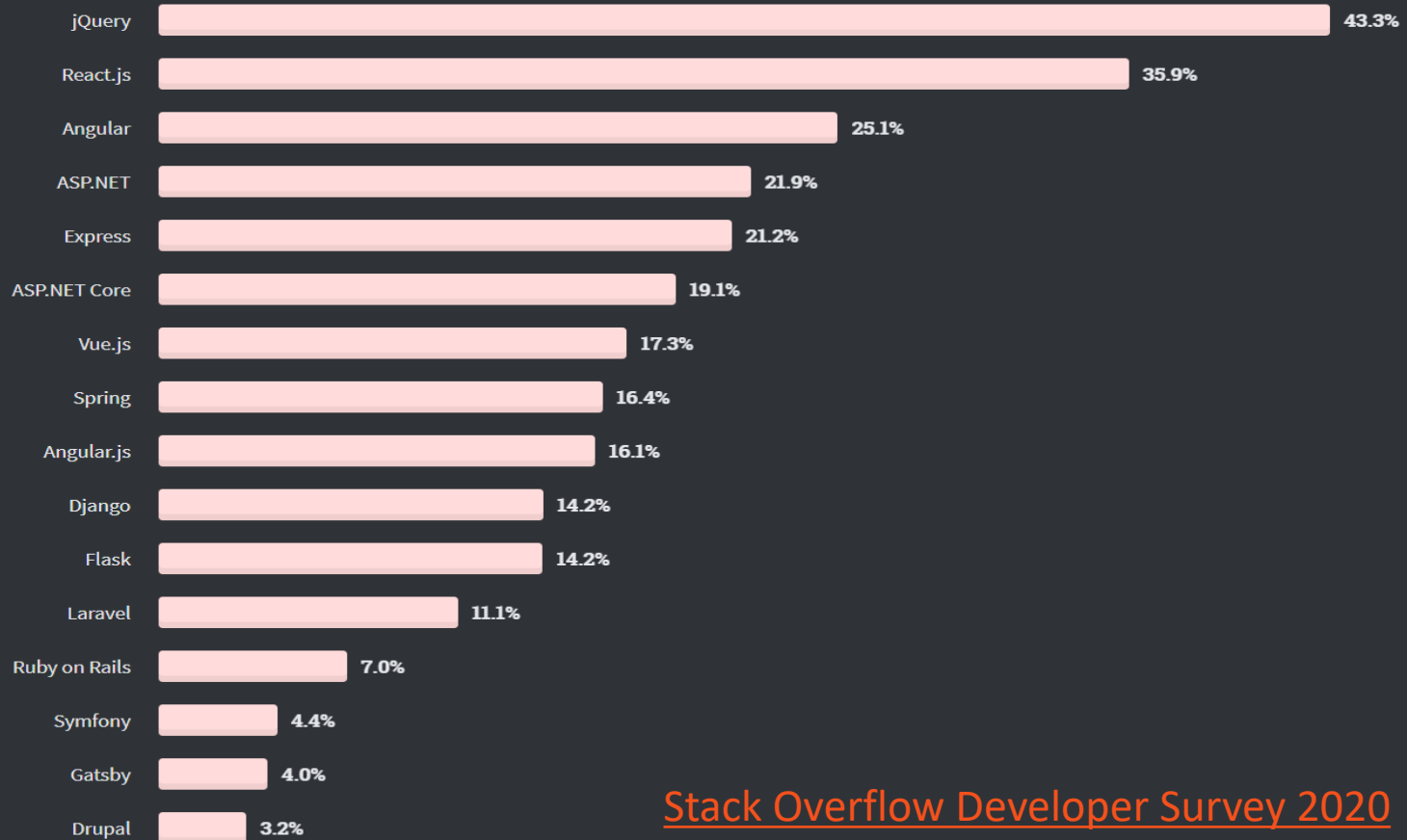
Web Servers Market share*

Product	Vendor	Platform	Percent
Apache	Apache	Open Source Cross Platform	34.5%
Nginx	Nginx, inc	Free + Paid versions Cross Platform	33.7%
Cloudflare Server	Cloudflare , inc	Growing fast!!	17.6%
IIS (Internet Information Services)	Microsoft	Windows, Included with Windows (Windows Server, Windows 10 Pro)	7.2%
LiteSpeed	LiteSpeed	Proprietary, Linux	8%
GWS (Google Web Server)	Google	Custom Linux-based Web server that Google uses for its online services	1.2%

*February 2021

https://en.wikipedia.org/wiki/Web_server

Popular Web Frameworks



[Stack Overflow Developer Survey 2020](#)

Database Platform?

Database Platforms



ORACLE®



Development/Programming Platforms?

Development/Programming Platforms



Microsoft®

.NET



Java™

You  <http://www.youtube.com/watch?v=8Px-GHPxB4I>

 3min

Cloud Computing/Cloud Platforms?

Cloud Computing/Cloud Platforms



They rent Cloud based services like Virtual Machines (Computers with OS running in the Cloud), Web Server, Database Systems to Customers based on Monthly Fees

Amazon Web Services



Atle Jørgensen er del av et lite utviklerteam i Sør-Afrika, som de siste 7 årene har utviklet skytjenester for Amazon. (Foto: Marius Jørgenrud)

Atle (42) står bak Amazon-skyen

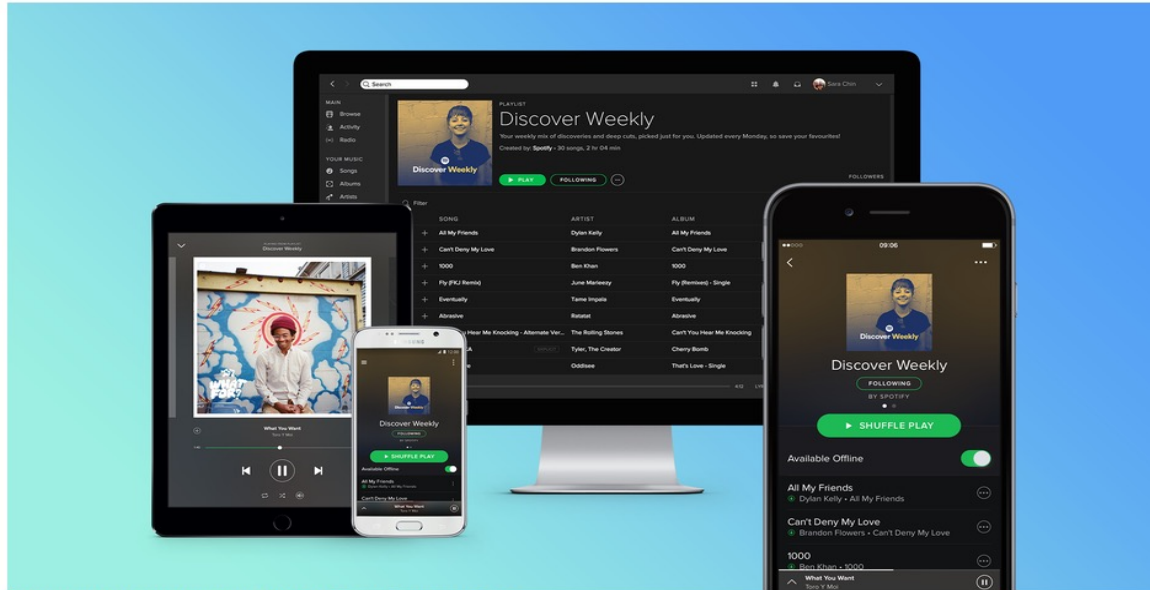
INTERVJUET: Fikk drømmejobben på surfetur i Sør-Afrika.

<http://www.digi.no/919886/atle-42-staar-bak-amazon-skyen>

Google Cloud Platform



Google Cloud Platform



Spotify's musikkstrømming skal heretter skje fra Googles nettsky. Foto: Spotify

GOOGLE CLOUD PLATFORM

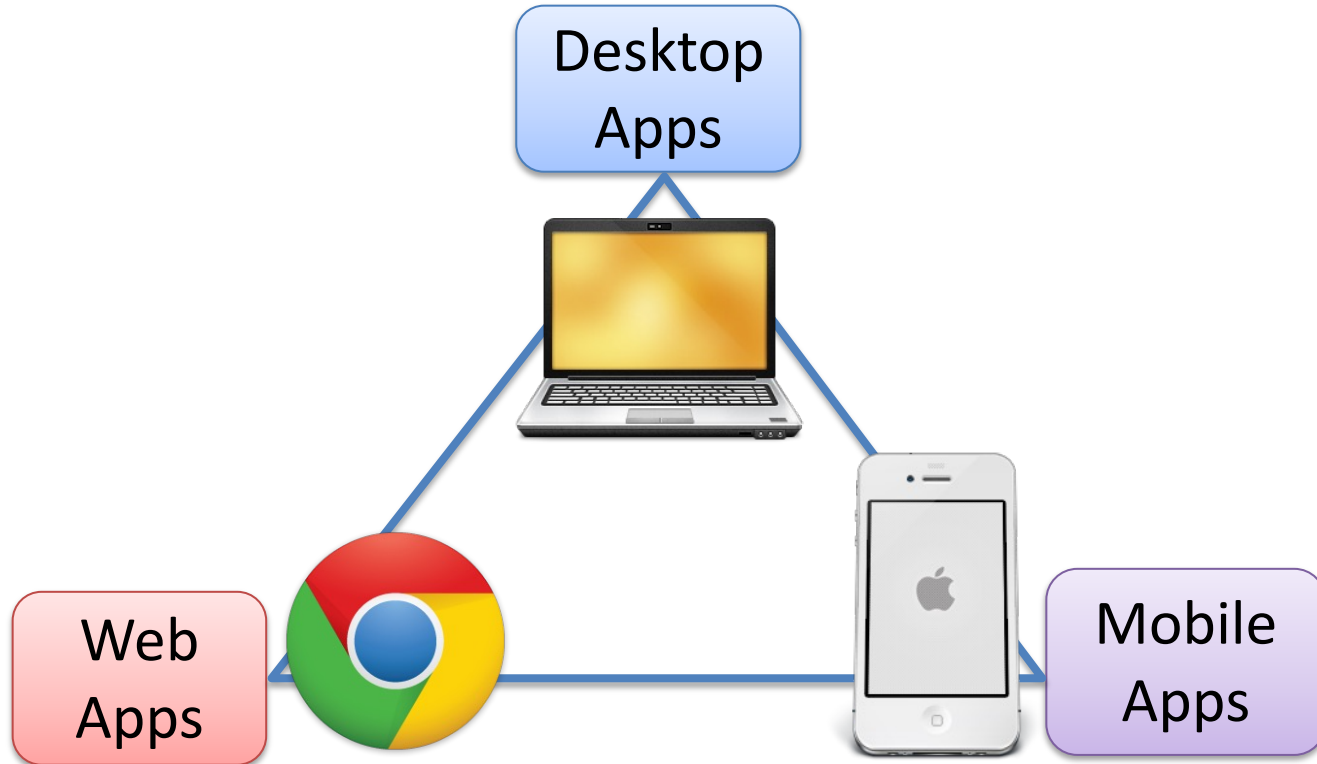
Spotify flytter til Google

Kvitter seg med egne datasentre.

<http://www.digi.no/bedriftsteknologi/2016/02/24/spotify-flytter-til-google>

Web vs. Desktop vs. Mobile

Advantages/Disadvantages?



Advantages/Disadvantages

Desktop

- Good Performance
- Different Platforms, different Programming methods and languages
- Installation is not always easy

Web

- Runs inside a Web Browser
- The Performance is not so good as it is for ordinary Desktop Applications
- Multiplatform, Works on all platforms, Code only once
- More complicated to create rich user interfaces
- Some differences in the behavior depending on the Web Browser.
- End-user don't need to install anything

Mobile Devices

- Good performance, but less performance than an ordinary computer
- Different Platforms, different Programming methods and languages
- You need to Create 3 different Apps (one for each device)
- Easy to install (App Store) and use Apps

Cloud Computing

- **SaaS** – Software as a Service
 - Software as a Service provides you with a completed product that is run and managed by the service provider.
 - You don't have to worry about the installation, setup and running of the application. Service provider will do that for you. You just have to pay and use it through some client.
 - Examples: Google Apps, Microsoft Office 365, web-based email systems
- **PaaS** – Platform as a Service
 - Providing a platform on which software can be developed and deployed.
 - Platforms as a service remove the need for organizations to manage the underlying infrastructure (usually hardware and operating systems) and allow you to focus on the deployment and management of your applications.
 - Examples: AWS, Microsoft Azure,... (e.g., use a preinstalled Web Server without worrying about anything else)
- **IaaS** – Infrastructure as a Service
 - Providing a full infrastructure in the cloud, such as Virtual Machines, Servers, OS, ...
 - Highest level of flexibility and management control over your IT resources and is most similar to existing IT resources that many IT departments and developers are familiar with today.
 - Examples: AWS, Microsoft Azure,...

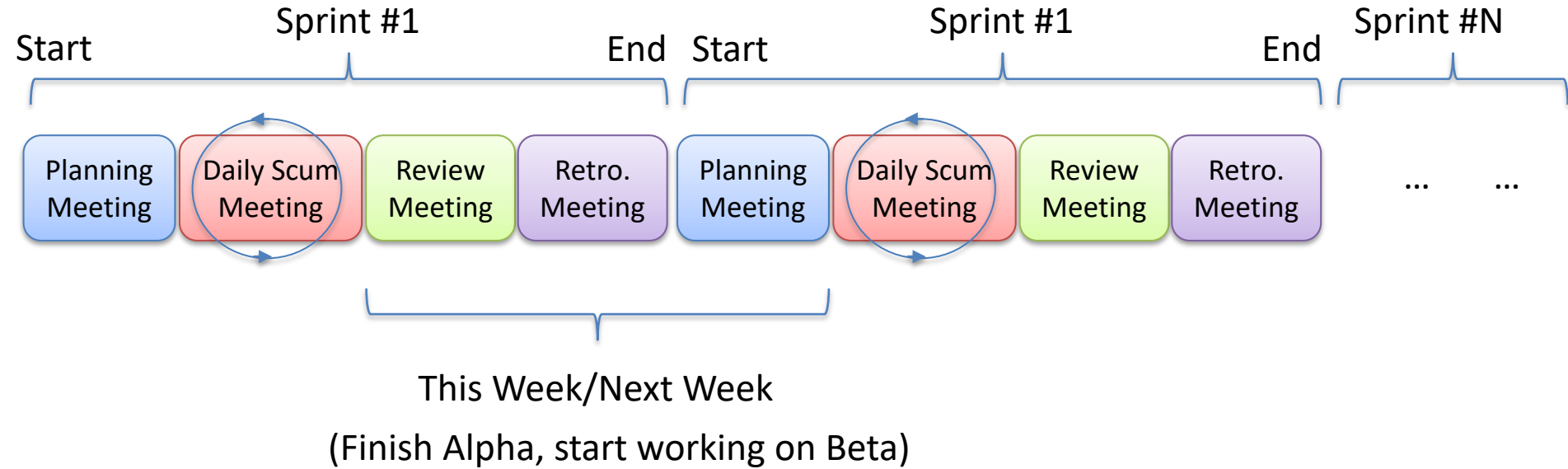


Scrum Meetings

4 different Scrum Meetings

- **Sprint Planning Meeting**
 - The purpose with the Sprint Planning is to discuss and select the work items for the next Sprint.
 - You select work items from the **Product Backlog** into the next **Sprint Backlog**.
- **Daily Scrum Meeting (Standup Meeting). 3 Questions:**
 1. What has been accomplished since last meeting?
 2. What will be done before the next meeting?
 3. What obstacles/impediments are in the way?
- **Sprint Review Meeting**
 - The purpose with the Sprint Review is to have a complete review of all the tasks/features that should be completed in the Sprint (Sprint Backlog items)
 - In this meeting, your team **demonstrates the features** that it completed in the sprint
- **Sprint Retrospective Meeting**
 - Learn from previous Sprints. Find Improvements the Scrum Team will agree on for the next Sprint.
 - Making Actionable Commitments
 1. Keep Doing
 2. Start Doing
 3. Stop Doing

Scrum Meetings



The **Scrum Master** is responsible for that these meetings are held



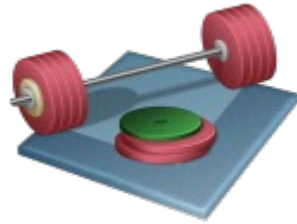
Web Site

Web Site

- **Create a Web Site/page for your Product**
 - Create HTML Web Site/page
 - Install on USN Web Server (Apache Web Server)
- Learn more about the Platforms (like Web Servers, etc.) you are using in your Project
 - Make sure you have Sketches and Descriptions of your Platforms in the SRS/SDD (->SRD) (“System Overview”)

See Next Slides for more details...

Create a Web Site for your Product



Start creating a Web Site with a short Presentation (Figures and Text) of your Software Product and with Documents available for download (Hyperlinks).

Include the Documents (PDF Files) created so far:

- Software Development Plan (SDP)
 - Software Requirements & Design (SRS/SDD -> SRD)
- (Other Documents and Resources should be added later)
- Link to Azure DevOps project

See Next Slides for more details...

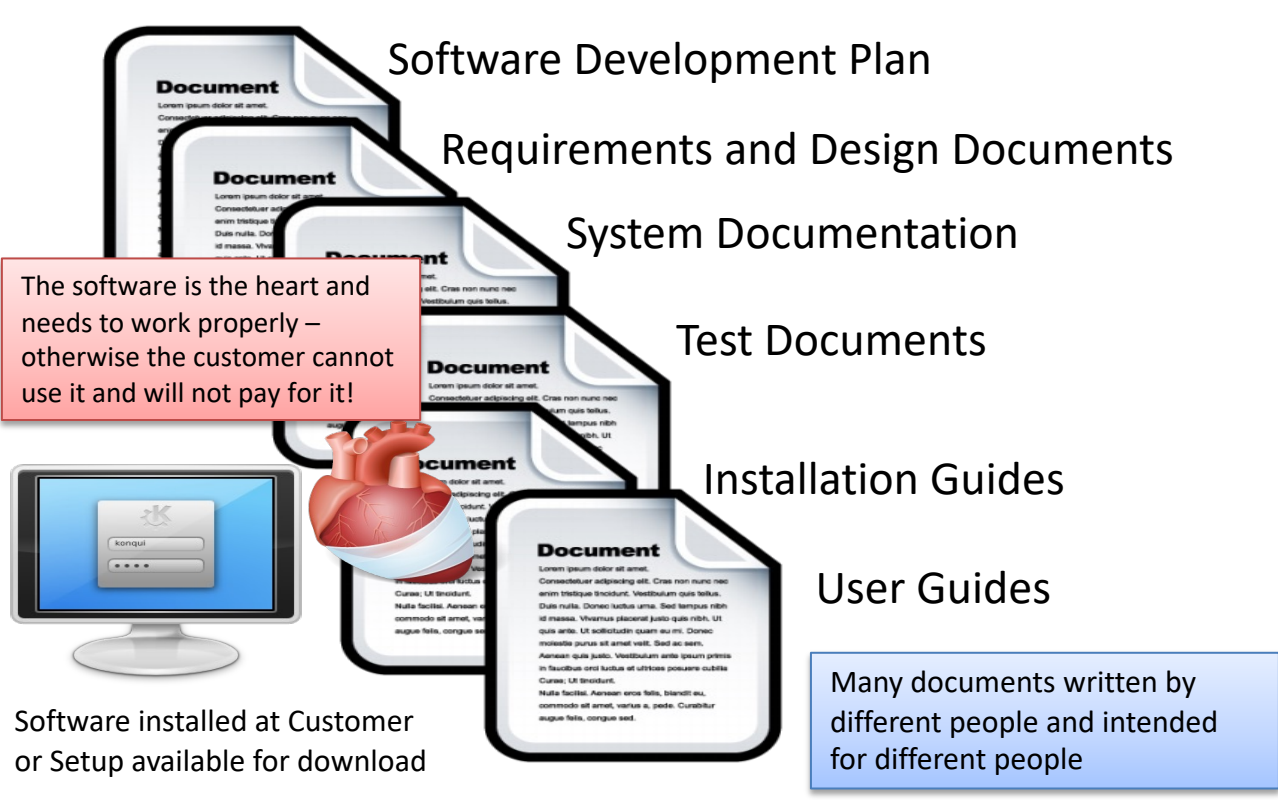
“Traditional School Project” vs. “Real Software Development Project”

One Report with “everything”



One document written by 3-6 students together

Does the system works?
Hopefully – but never used or tested by the reader (or the students?)



The software is the heart and needs to work properly – otherwise the customer cannot use it and will not pay for it!



Software installed at Customer or Setup available for download

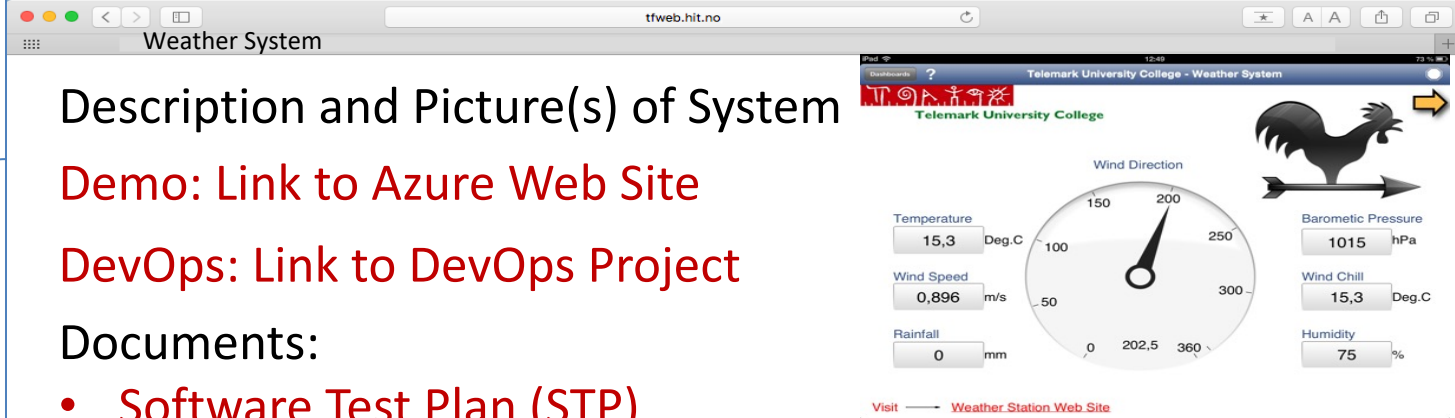
Many documents written by different people and intended for different people

“Final Report”

Note! The Web Site you create now will be the first draft of the “Final Report” for your Project.
The best Web Site in each Team will be selected and used further in your Project

“Chapters” and “Appendices” in the “Report”

Note! For “Read-only” Documents: Use PDF Format



Description and Picture(s) of System

Demo: [Link to Azure Web Site](#)

DevOps: [Link to DevOps Project](#)

Documents:

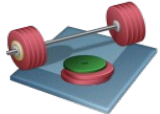
- [Software Test Plan \(STP\)](#)
- [Software Requirements and Design \(SRD\)](#)
- [Software Test Plan \(STP\) + Test Documents](#)
- [System Documentation](#)
- [User Guide\(s\)](#)
- [Installation Guide\(s\)](#)
- ...

Files:

- [InstallationFiles.zip](#) Everything you have created in Azure
- [AzureDevOps.zip](#) DevOps (Code, Original Documents, etc..)

Files needed to install the System.
How to install is explained in the Installation Guide(s)

Web Site Example



Create a similar Web Site for your Product

Presentation (Figures and Text) of your Software Product

Note! We should use only HTML (not ASP.NET! – because the Web Server we are using at school don't support it). In addition you may use JavaScript, CSS, PHP

Documents available for download (Hyperlinks)

Weather System

The Weather System ...

ambient weather

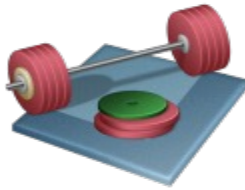
6:25 PM 47% 5.4TH 4.12 inch 31.27 82.2°F 89% 27.0°F 51%

Documents

- Software Development Plan
- Software Requirements & Design

Note! Always use PDF Files!!!

Web Server



Deploy your Web Site
according to this information

We will use the available Web Server at the university. We have 2 different options. Select one of the following:

- **web01.usn.no:** Running Linux. Web Server: Apache
- ~~**home.usn.no:** Running Windows. Web Server: Microsoft Internet Information Server – IIS~~
(No longer available for new users)

https://web01.usn.no

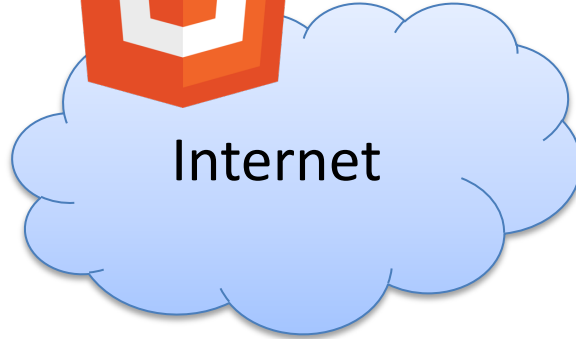
Server-side



HTML



CSS



Clients



PCs with Web Browsers

Apache

HTTP SERVER



https://web01.usn.no

- Server
 - Operating System: Linux
 - Web Server: Apache
 - Database: MySQL
 - Supported Languages: HTML, CSS, PHP
- Example: <https://web01.usn.no/~hansha/>

https://web01.usn.no

- <https://web01.usn.no/~username>
- User Name = Student Number
- Allowed Start Pages:
 - **index.html**
- **FTP: WinSCP**
 - Host Name: web01.usn.no, User Name: Your Student Number
 - Create a folder **public_html** on the server where you put your HTML files inside
 - Note! Linux distinguishes between uppercase and lowercase letters (use index.html NOT Index.html)
- <https://min.usn.no/programvare/eget-webomrade-web01-usn-no-article217606-32428.html>

Note! Set proper Read/Write Properties for both your root folder and for the public_html folder by right-clicking and select Properties on these folders in your FTP program

https://web01.usn.no

Not Working Checklist:

- Use correct Url: <https://web01.usn.no/~username>
- Use **index.html** NOT Index.html (Why?)
- Create and put your .html files into a folder **public_html** on the Web Server
- Set proper Read/Write Properties for the public_html folder by right-clicking and select Properties in the FTP program
- Set proper Read/Write Properties for your root folder by right-clicking and select Properties in the FTP program. The name of the root folder is your Student Number

Web Site

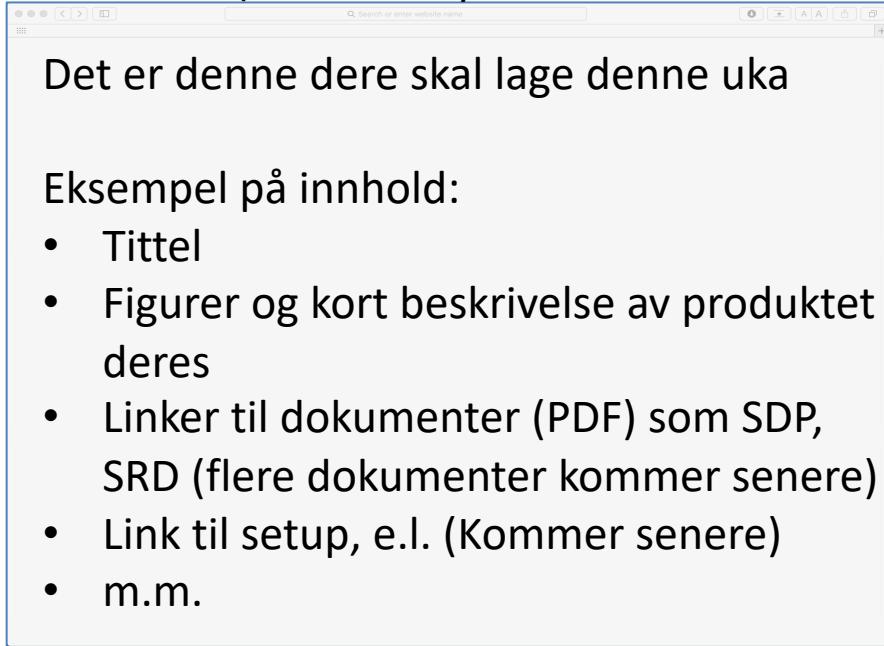
- We shall create a Web Site (HTML, CSS) that introduces the product you are creating, including documents you have created (SDP, SRD, ...)
- This is an individual activity, meaning all team members should create such a web site.
- You need to know basic **HTML**. A good source is: <https://www.w3schools.com/html/>
- Recommended HTML Editor: **Visual Studio Code** (or you can use Visual Studio, but VS is not well suited for HTML pages)
- The best web site within each group should then be selected and used further in the project.
- We will use the available Web Server at the university. The Web address (URL) will be like this: <https://web01.usn.no/~hansha> (You should use your Student Number)

Resources:

<https://min.usn.no/programvare/eget-webomrade-web01-usn-no-article217606-32428.html>

Web Development in your Project

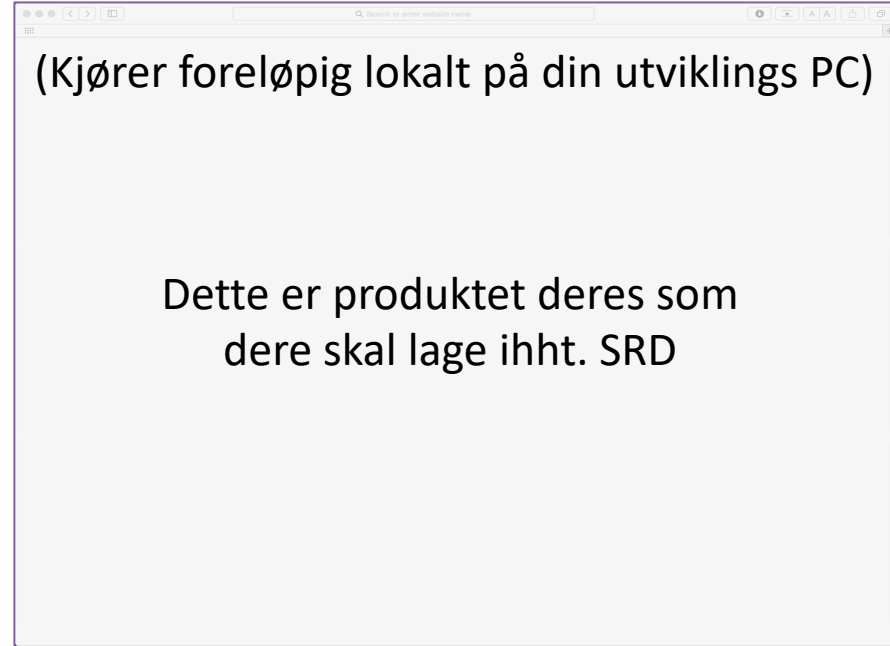
1 HTML (Installeres på USN sin Web Server)



“Commercial” Web Page with information about your Product, you can download setup files, documents, etc.

ASP.NET

2



Your web-based Product -> Web Application

vs.

Illustrative Example

1

The Vendors Home Page

The screenshot shows the Canvas website home page. At the top, there is a navigation bar with 'United Kingdom', a search icon, and links for 'For Business', 'Support', 'Contact Us', 'Login', and 'Account'. Below this is the 'INSTRUCTURE' logo and a menu with 'SCHOOLS', 'HIGHER & FURTHER EDUCATION', 'RESOURCES', 'NEWS & EVENTS', 'ABOUT US', and a 'GET IN TOUCH' button. The main content area features a large banner for 'CANVAS Teaching and learning. Powered by Canvas.' with a video player showing a woman working on a laptop. Below the banner is a 'LEARN ABOUT OUR PLATFORM' button. On the right side, there is a vertical sidebar with icons for 'Dashboard', 'Courses', 'Calendar', 'Inbox', 'History', 'Commons', and 'Help'.

Product Information Web Site

2

Canvas Web Application

The screenshot shows the Canvas web application dashboard. At the top, there is a 'Dashboard' header and a 'To Do' list on the right. The main content area is titled 'Published Courses (7)' and displays a grid of course cards. Each card has a cover image and a title. The 'To Do' list on the right contains several items with status indicators and due dates. At the bottom right, there is a 'Recent Feedback' section and a 'View Grades' button.

Dashboard

Published Courses (7)

- Grade Data Communication IIA2017-1 21V Industrial Informati 0 points • Feb 4 at 8am
- Grade UML IIA412-1 21V Systemutvikling 0 points • Feb 16 at 10:15am
- Grade OPC Lab Work IIA2017-1 21V Industrial Informati 0 points • Feb 21 at 11:59am
- Grade Innlevering viteskapsteori PRH612-1 21V Bacheloroppgave 0 points • Feb 28 at 11:59am
- Grade Egenerklæringskema / Self declaration from EIK-001-1 2018 Skoleåret 1 point • No Due Date

Coming Up View Calendar

- Lab Industrial Information Technology IIA2017-1 21V Industrial Informati Feb 22 at 1:15pm
- Forelesning Systemutvikling og dokumentasjon IIA412-1 21V Systemutvikling Feb 23 at 10:15am
- Forelesning Industrial Information Technology IIA2017-1 21V Industrial Informati Feb 24 at 9:15am

3 more in the next week ...

Recent Feedback
Nothing for now

View Grades

Product Web Application

vs.

WYSIWYG HTML Editors

- **Adobe Dreamweaver** (Monthly Payment)
- Kompozer (Free)
- Bluegriffon (Free)
- Sparkle (\$80), etc...

WYSIWYG – What You See Is What You Get
You don't need to know HTML syntax - Its just like using MS Word.

Other HTML Editors (not WYSIWYG)

- **Visual Studio Code**
- Visual Studio (ASP.NET)
- CoffeeCup (\$69, Free Trial)
- Coda (\$99)
- ... NotePad

Only possible to change the HTML source code and then select “Preview” in order to see how it looks like in a Web Browser.
You need to know HTML syntax

Simple Web Page in Visual Studio Code

The image shows the Visual Studio Code editor with a file named 'webpage.html' open. The code is as follows:

```
1 <!DOCTYPE html>
2 <html>
3
4 <body>
5
6   <h1>Heading 1</h1>
7   <h2>Heading 2</h2>
8   <h3>Heading 3</h3>
9
10  <p>My first paragraph.</p>
11
12  <p>
13    <a href="https://www.w3schools.com">This is a link</a>
14  </p>
15
16  <p>
17    
18  </p>
19
20  <p>
21    
22  </p>
23
24 </body>
25
26 </html>
```

The rendered output in the browser shows the following content:

Heading 1
Heading 2
Heading 3
My first paragraph.
[This is a link](https://www.w3schools.com)

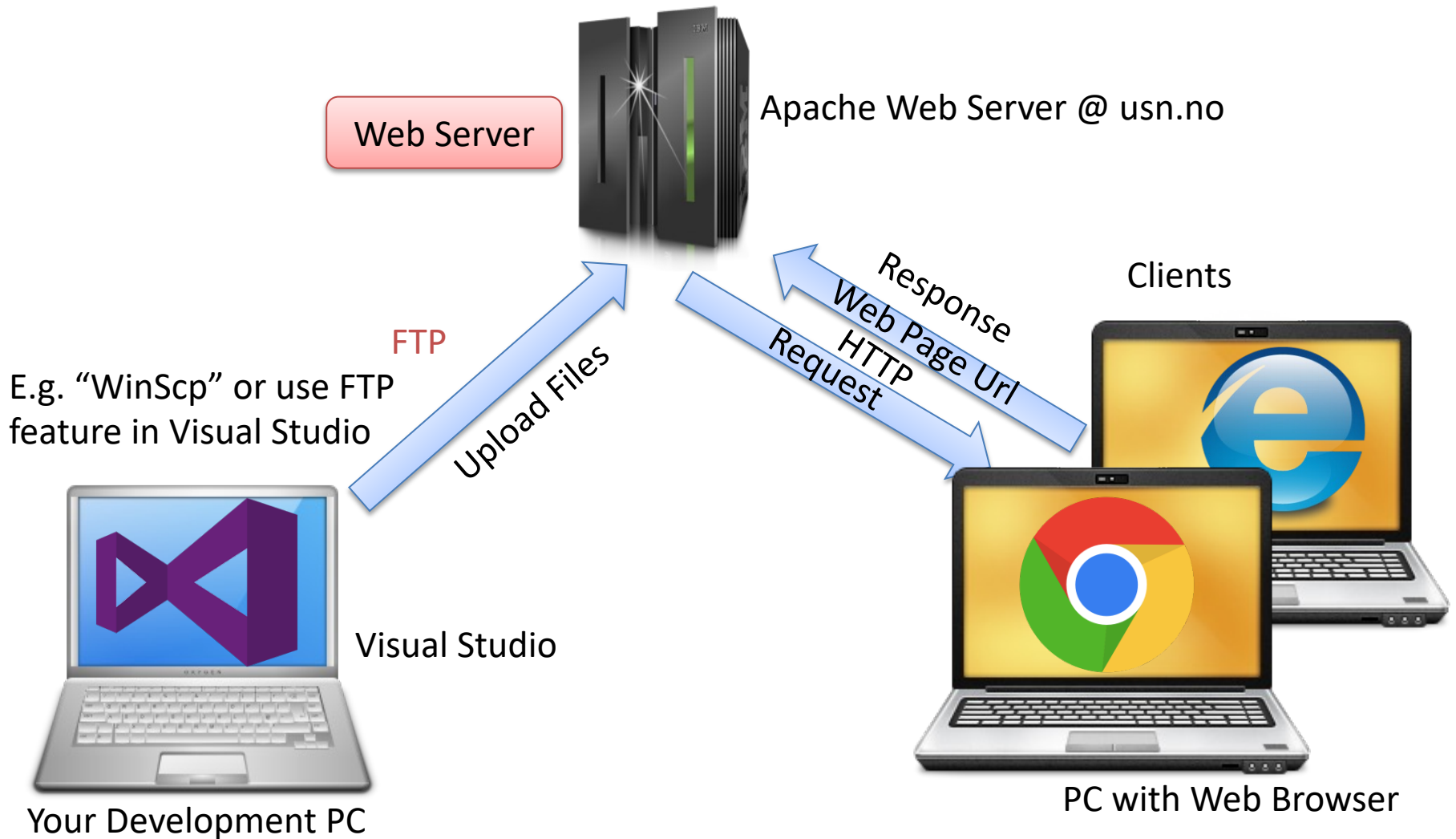
The image also shows a preview of a weather system interface with a smartphone displaying weather data and a weather station.

Documents

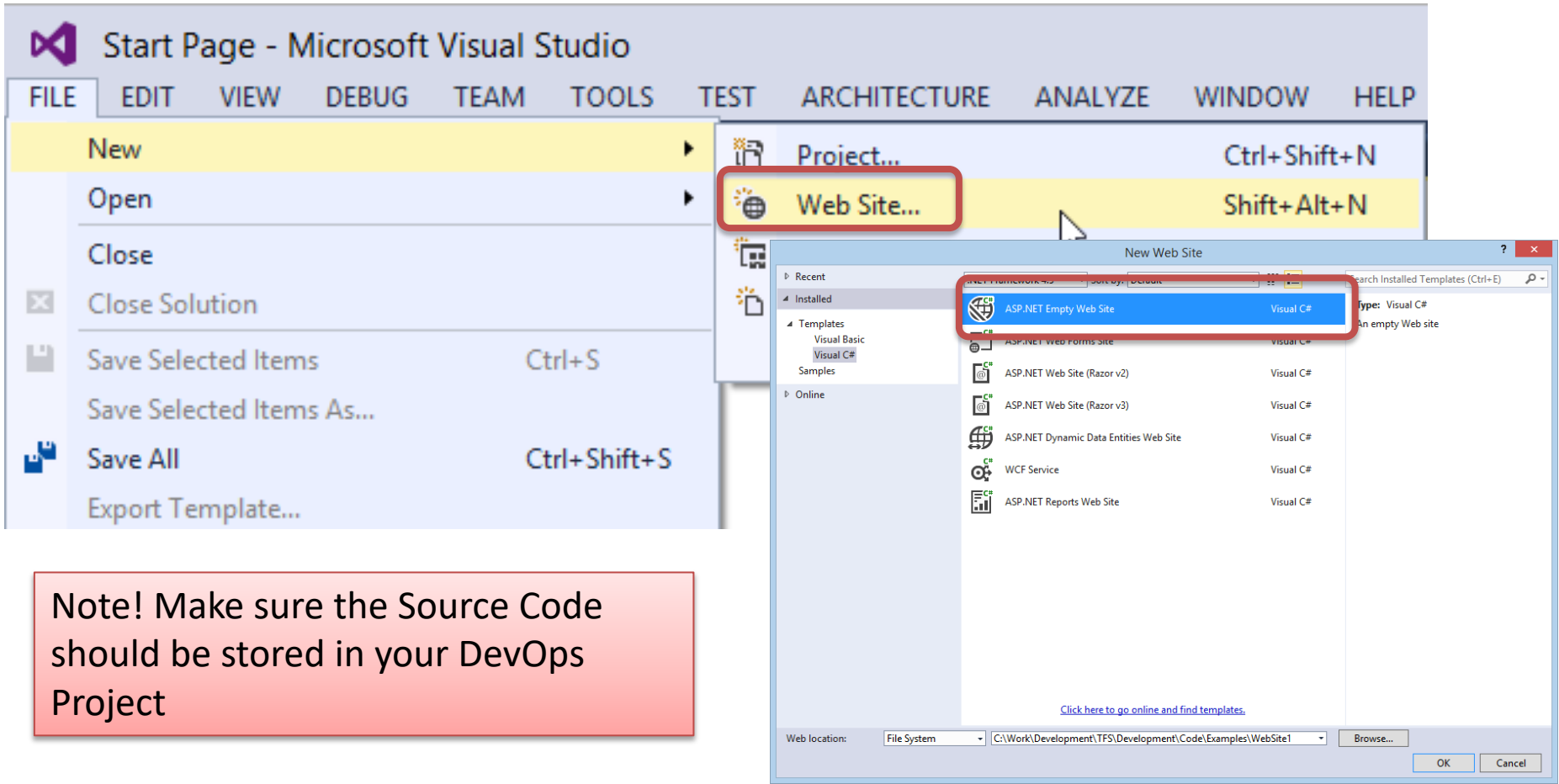
- Software Development Plan
- Software Requirements & Design

Create a Web Site with Visual Studio

- Visual Studio is not well suited for creating Static HTML Web Pages.
- Visual Studio has poor WYSIWYG Editing possibilities
- We can use Visual Studio because we already use it in our Project – and basic HTML syntax is something you should know about.
- Visual Studio is more optimized for creating Dynamic Web Pages and creating ASP.NET Web Pages in special
- I recommend that you use **Visual Studio Code** instead



Create a Web Site with Visual Studio



The image shows the Visual Studio Start Page with the 'New' menu open. The 'Web Site...' option is highlighted in yellow and has a red box around it. Below it, the 'New Web Site' dialog box is open, showing a list of templates. The 'ASP.NET Empty Web Site' template is selected and highlighted in blue, with a red box around it. The dialog box also shows the 'Web location' field set to 'File System' and the path 'C:\Work\Development\TFS\Development\Code\Examples\WebSite1'.

Start Page - Microsoft Visual Studio

FILE EDIT VIEW DEBUG TEAM TOOLS TEST ARCHITECTURE ANALYZE WINDOW HELP

New Project... Ctrl+Shift+N

Web Site... Shift+Alt+N

Close Solution

Save Selected Items Ctrl+S

Save Selected Items As...

Save All Ctrl+Shift+S

Export Template...

New Web Site

ASP.NET Empty Web Site Visual C#

ASP.NET Web Forms Site Visual C#

ASP.NET Web Site (Razor v2) Visual C#

ASP.NET Web Site (Razor v3) Visual C#

ASP.NET Dynamic Data Entities Web Site Visual C#

WCF Service Visual C#

ASP.NET Reports Web Site Visual C#

Web location: File System C:\Work\Development\TFS\Development\Code\Examples\WebSite1

OK Cancel

Note! Make sure the Source Code should be stored in your DevOps Project

Create a Web Site with Visual Studio

The image shows a screenshot of Visual Studio with the 'Add New Item' dialog box open. The dialog is titled 'Add New Item - MyWebSite' and shows a list of installed templates. The 'HTML Page' template is selected and highlighted with a red box. The 'Name' field at the bottom of the dialog is also highlighted with a red box and contains the text 'index.htm'. A blue callout box with white text says 'Your Start Page needs to be named "index.htm"'. The 'Add' button is visible at the bottom right of the dialog.

Visual Studio Add New Item Dialog:

- Dialog Title: Add New Item - MyWebSite
- Search Installed Templates (Ctrl+E): [Search Box]
- Type: Visual C#
- Description: An HTML page that can include client-side code
- Installed Templates List:

Template Name	Type
HTML Page	Visual C#
JavaScript File	Visual C#
Style Sheet	Visual C#
Web Form	Visual C#
Master Page	Visual C#
Web User Control	Visual C#
ADO.NET Entity Data Model	Visual C#
Browser File	Visual C#
Class	Visual C#
Class Diagram	Visual C#
CoffeeScript	Visual C#
DataSet	Visual C#
- Name: index.htm
- Buttons: Add, Cancel

Visual Studio 'Add' Menu:

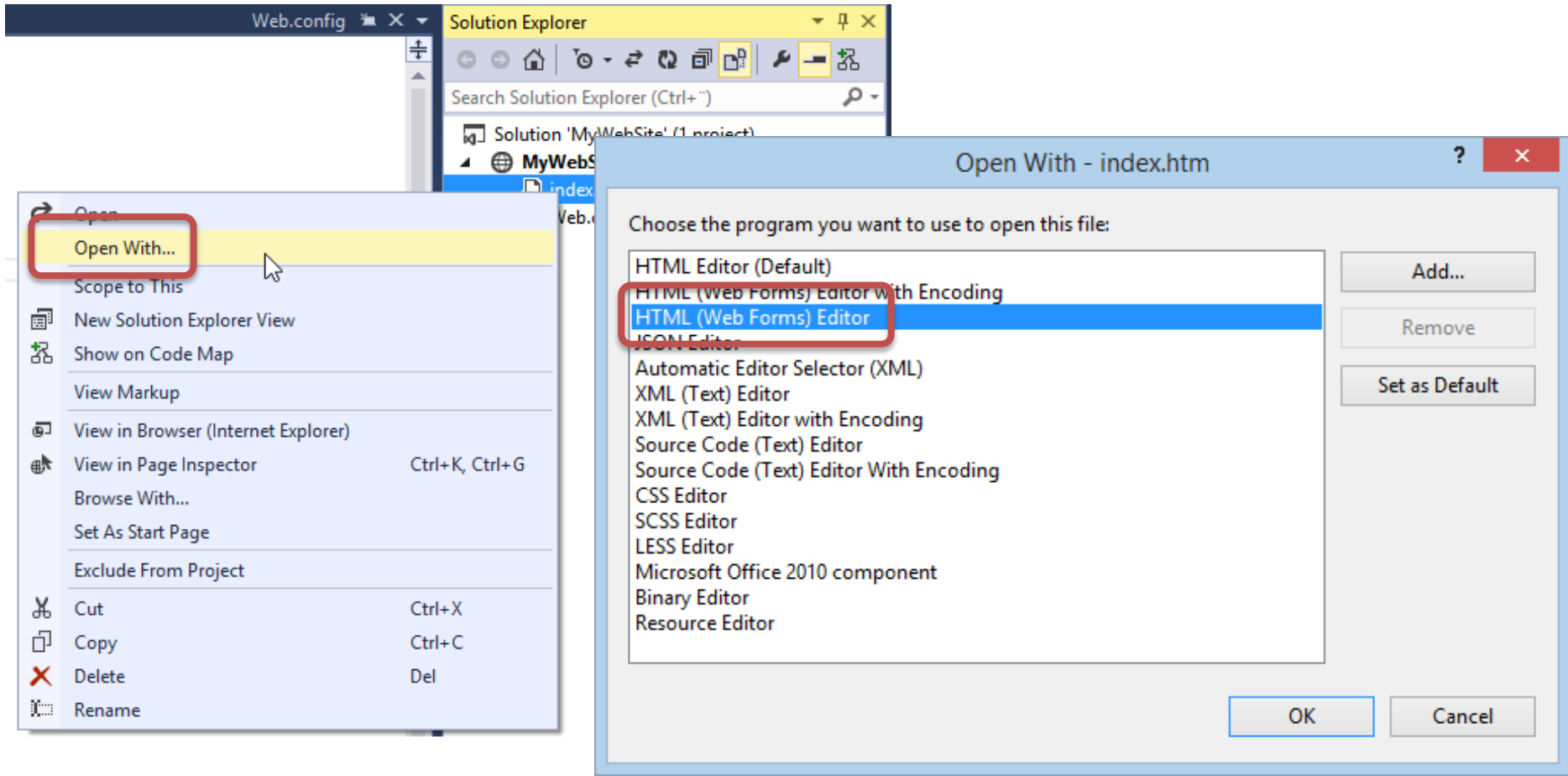
- Build Web Site (Shift+F6)
- Publish Web Site
- Scope to This
- New Solution Explorer View
- Show on Code Map
- Add
 - View Class Diagram
 - Manage NuGet Packages...
 - Copy Web Site...
 - Start Options...
 - Set as StartUp Project
 - View in Browser (Internet Explorer)
 - View in Page Inspector (Ctrl+K, Ctrl+G)
 - Browse With...
 - Refresh Folder
 - Source Control
- Cut (Ctrl+X)
- Copy (Ctrl+C)
- Paste (Ctrl+V)
- Remove (Del)
- Open Folder in File Explorer
- Properties Window (Ctrl+W, P)
- Property Pages (Shift+F4)
- Run Code Analysis on Web Site

Visual Studio HTML Editor

Note! Due to the brand new HTML editor in Visual Studio, static HTML files no longer has the Design | Split | Source options enabled by default.

- That's because the new HTML editor is the default editor for all HTML files with the exception of ASP.NET Web Forms files (.aspx, .ascx, .master).
- However, it is only the Web Forms editor that has support for the designer and split view.
- So all we have to do is to map our .html or .htm files to use the Web Forms editor instead of the new HTML editor.
- Simply right-click any .html/.htm file in Solution Explorer and select Open With...

HTML Editor in Visual Studio



Toolbox

Search Toolbox

- HTML
 - Pointer
 - Input (Button)
 - Input (Reset)
 - Input (Submit)
 - Input (Text)
 - Input (File)
 - Input (Password)
 - Input (Checkbox)
 - Input (Radio)
 - Input (Hidden)
 - Textarea
 - Table
 - Image
 - Select
 - Horizontal Rule
 - Div
- General

There are no usable controls in this group. Drag an item onto this text to add it to the toolbox.

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
  <title></title>
</head>
<body>
</body>
</html>
```

Create your HTML Code here!

Code Editor

100 %

body

Simple "WYSIWYG" Editor + Preview

Solution Explorer

Search Solution Explorer (Ctrl+')

- Solution 'MyWebSite' (1 project)
 - MyWebSite
 - index.htm
 - Web.config

Solution Explorer Team Explorer

Properties

DOCUMENT

Charset	
Class	
Id	
Style	
Title	

Charset
Character set used to encode the document.

```
<!DOCTYPE html>
<html>
<body>

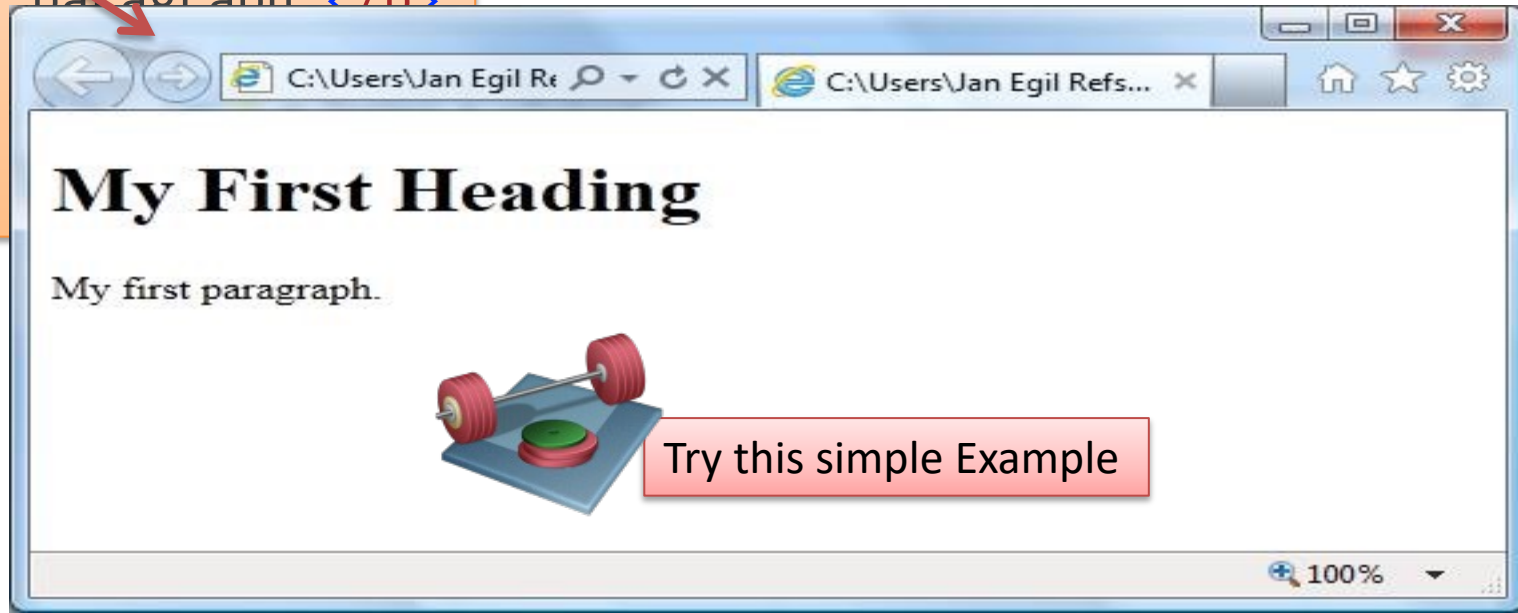
<h1>My First Heading</h1>

<p>My first paragraph </p>

</body>
</html>
```

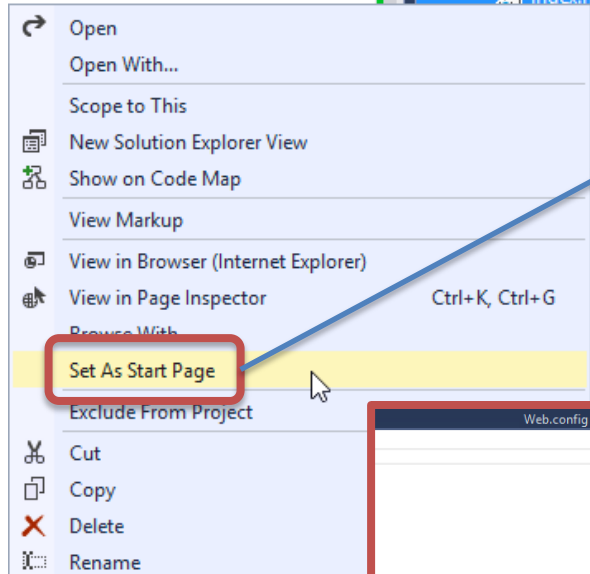
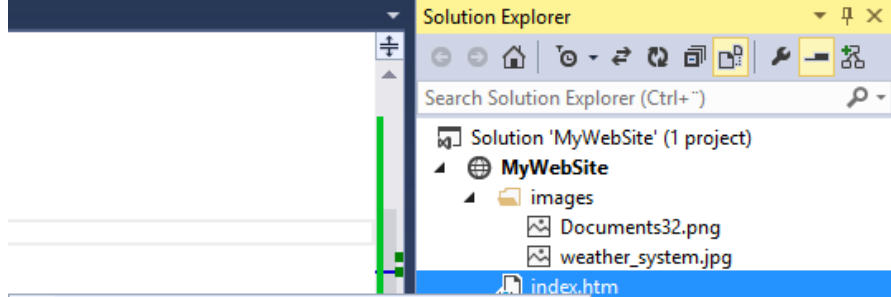
HTML Code

Simple HTML Page Example

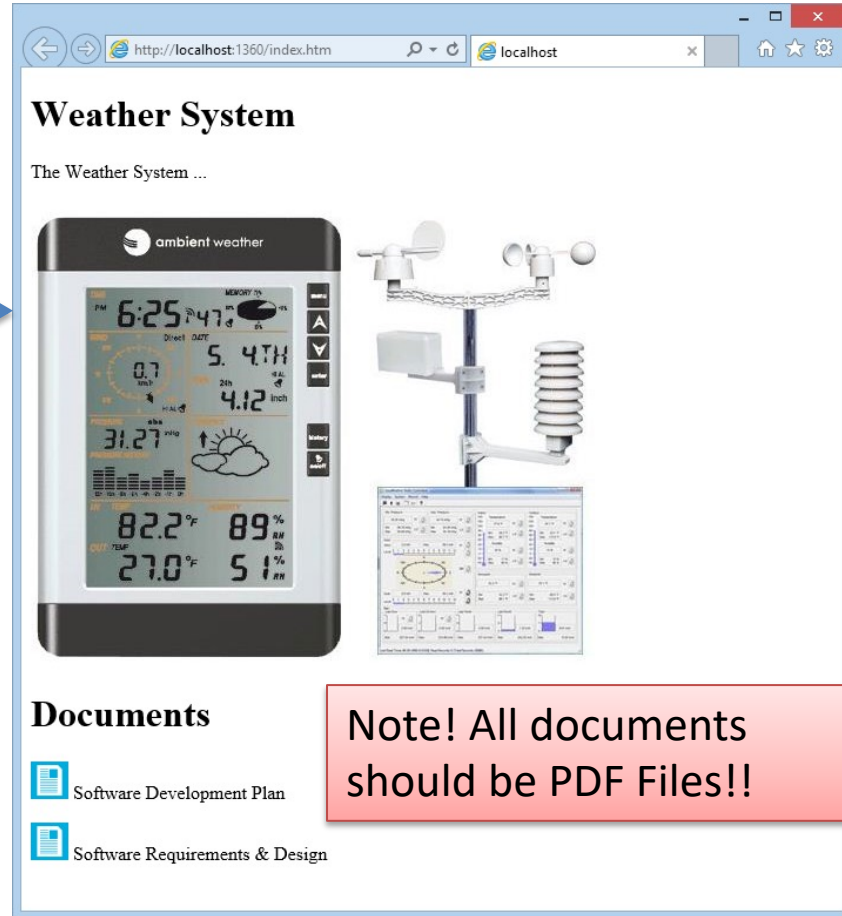


Web Browser:

Test in Browser

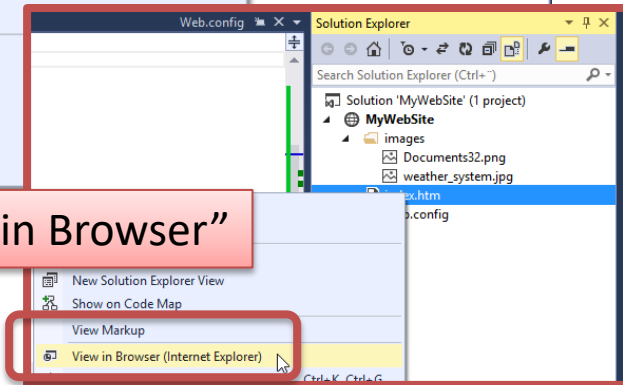


Hit F5 to Run It

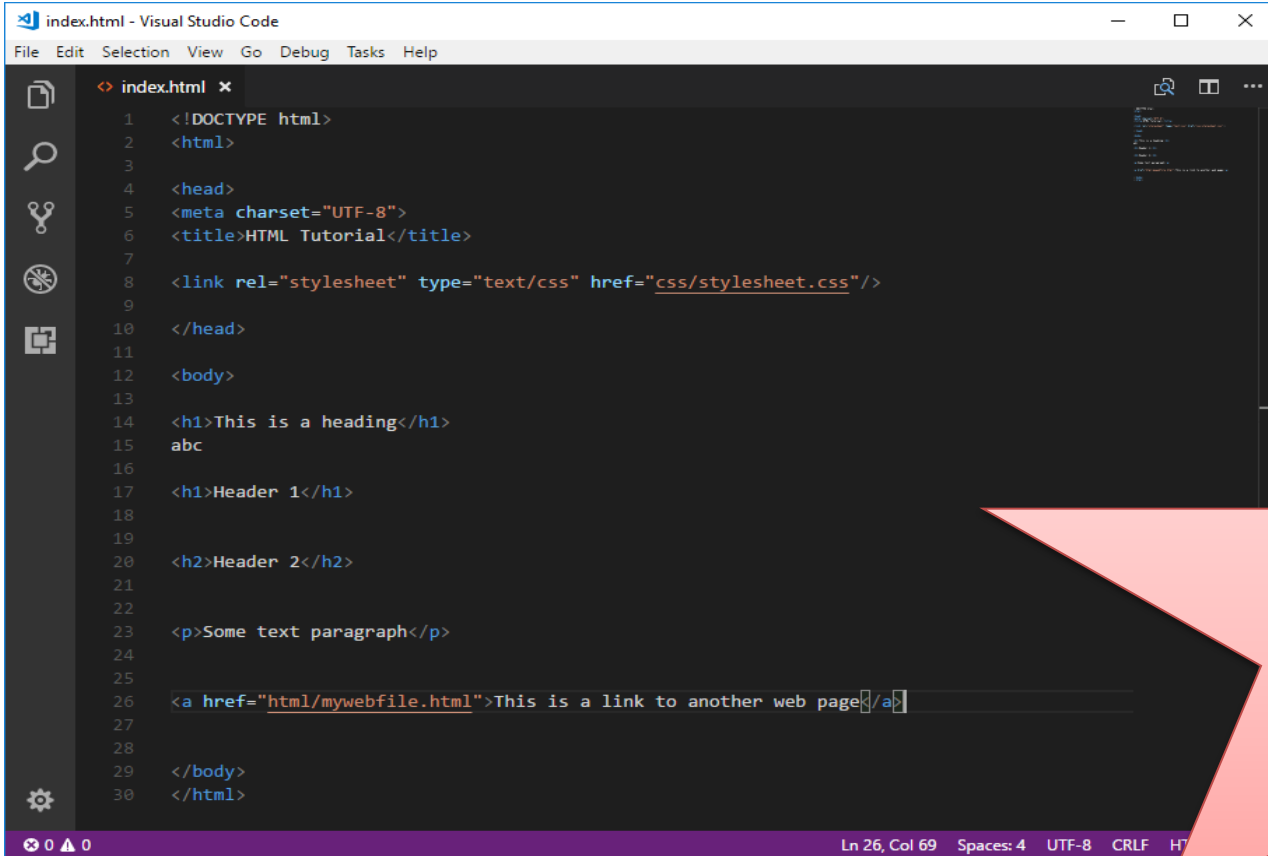


Note! All documents should be PDF Files!!

Or Select "View in Browser"




Visual Studio Code



```
index.html x
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5 <meta charset="UTF-8">
6 <title>HTML Tutorial</title>
7
8 <link rel="stylesheet" type="text/css" href="css/stylesheet.css"/>
9
10 </head>
11
12 <body>
13
14 <h1>This is a heading</h1>
15 abc
16
17 <h1>Header 1</h1>
18
19
20 <h2>Header 2</h2>
21
22
23 <p>Some text paragraph</p>
24
25
26 <a href="html/mywebfile.html">This is a link to another web page</a>
27
28
29 </body>
30 </html>
```

Ln 26, Col 69 Spaces: 4 UTF-8 CRLF HT

Visual Studio Code is a simple, lightweight and easy to use cross-platform Editor



Recommended

Download: <https://code.visualstudio.com/>

Hyper- links:

```
<!DOCTYPE html>
<html>
<body>

<h1>This is a heading</h1>

<a href="http://www.google.com">This is a link to Google</a>

</body>
</html>
```

Images:

```
<!DOCTYPE html>
<html>
<body>

<h1>This is a heading</h1>



</body>
</html>
```


HTML Tags

Hyperlink:

```
<a href="http://www.google.com">This is a link to Google</a>
```

Bold Text:

```
<b>This is my Text</b>
```

Paragraph:

```
<p>My first paragraph.</p>
```

Headers:

```
<h1>This is my Header</h1>
```

```
<h2>This is my Header</h2>
```

```
<h3>This is my Header</h3>
```

Line Break:

```
This is my Text
```

```
<br>
```

```
This is also my Text
```

Title:

```
<title>This is my Title</title>
```

Comments:

```
<!-- Write your comments here -->
```

Image:

```

```

HTML Tags - Table

```
<table width="200" border="1">
  <tr>
    <td>a</td>
    <td>b</td>
    <td>c</td>
    <td>d</td>
  </tr>
  <tr>
    <td>e</td>
    <td>f</td>
    <td>g</td>
    <td>h</td>
  </tr>
  <tr>
    <td>i</td>
    <td>j</td>
    <td>k</td>
    <td>l</td>
  </tr>
</table>
```

a	b	c	d
e	f	g	h
i	j	k	l

CSS (Cascading Style Sheet)

CSS is a stylesheet language that describes the presentation of an HTML page.

stylesheet.css:

```
body {  
    background-color: #d0e4fe;  
}  
  
h1 {  
    color: orange;  
    text-align: center;  
}  
  
p {  
    font-family: "Times New Roman";  
    font-size: 20px;  
}
```

myfile.htm:

```
...  
<head  
...  
    <link rel="stylesheet" type="text/css" href="stylesheet.css"  
/>  
...  
</head>  
...
```

HTML

Good Resource for creating Web Pages with HTML, CSS, JavaScript, SQL, etc.

<http://www.w3schools.com>

HTML: <https://www.w3schools.com/html/>

CSS: <https://www.w3schools.com/css/>

HTML Example



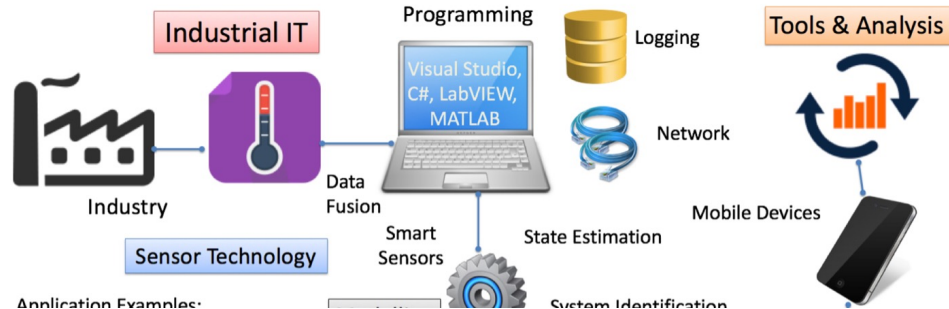
Industrial IT and Automation

© Published 2018.01.19 - Hans-Petter Halvorsen - Updated 2018.02.02 - Hans-Petter Halvorsen

Industrial IT Automation Programming

Industrial IT and Automation are important fields today and will be even more important in the future with the new age of digitalization, Internet of Things, data security and the industrial and environmental challenges of tomorrow. Industrial IT and Automation are needed in all kind of industries today and in the future. Industrial IT and Automation are knowledge for the future. The future is now - Let's change it together!

Do you want to study [Industrial IT and Automation?](#)



- Programming Languages: PHP, HTML5, CSS, JavaScript
- CSS: Bootstrap
- Database: MariaDB
- Server: Linux
- Web Server: Apache
- Responsive Web Design: The web pages look good on all devices (desktops, tablets, and phones).

<https://www.halvorsen.blog>



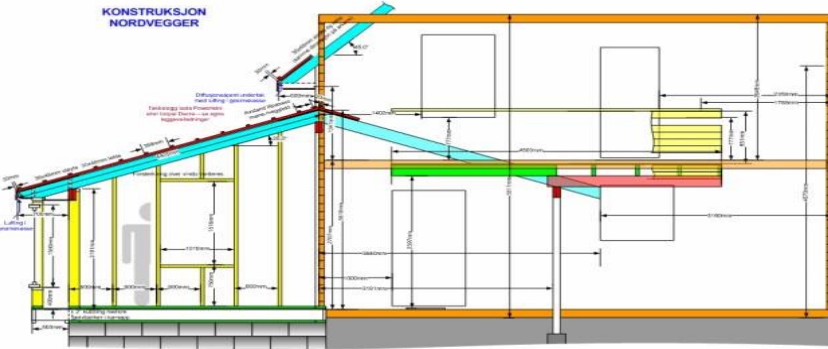
Sprint Review

Sprint Review

- Short Review of the “Alpha” Iteration.
- You should have a detailed Sprint Review session (“Alpha Release”) within your Team
 - Go through everything you have done in detail
- Sprint Retrospective
 - Inspect (Sprint Retrospective) your own Team (have a discussions) and try to figure out what was good and what that can be improved (your roles, how you work together, etc.)
- Update the Product Backlog with new Items (“User Stories”) based on the Review/Feedback

See Next Slides for more details...

Requirements/Design



Plans made and approved

Alpha



Foundation finished, building structure started
A "proof" that you can do it, PoC (Proof of Concept)

Beta



Building structure finished,
Inside work on track

RC



Furniture, Flowers and
small adjustments missing

RTM



Ready for Sale or Move in

Your “Alpha” Release/Iteration

1. Documents: The following Documents should be “finished”:

- Software Development Plan (SDP)**
 - Product Description, Team Description and Project Organization, Gantt Chart, Tools and Templates, etc.
- Software Requirements and Design (SRD)**
 - System overview, System and User Requirements, **Functional and Non-Functional Requirements**, **GUI Sketches**, **ER diagram**, **UML Diagrams**
- SDP and SRD can easily be found in Teams and you have made a good folder structure

2. Software/Programming:

- Tables Implemented in **SQL Server** (on everybody's Development PCs)– Generated from ERwin Script
- You know how to **Communicate with your SQL Server Database from C# Code** (You have made a small examples to for storing and reading data from the database used in your project)
- You have made some simple **ASP.NET** Example Applications (“Hello World App”, basic CRUD App, etc.)
- You have created **Classes** and **Methods** (according to the UML diagrams) - not the contents in the Methods, just the Names/declarations
- ALL Team Members have installed MS Project, erwin, SQL Server, Visual Studio, UML software (needed in project and exam!)
- ALL Team Members have started to do some Programming in Visual Studio!
- ALL Team Members have been given a clear responsibility (e.g., a separate module or application) when it comes to Programming!

3. Azure DevOps (give access to HPH):

- A good Folder structure has been made + Iterations (Alpha, Beta, RC, RTM)
- Product Backlog** has been made (A List of Requirements for your System)
- ER Diagram**, **Database Scripts**, etc. have been uploaded/Checked-in
- Code Examples/Prototypes have been uploaded/Checked-in + Same with Database Script

4. Project Web Page

- A simple **HTML Web Page** with an Introduction (Text and Figures) + Links to SDP and SRD (PDF) +++

Sprint Review Meeting



Purpose: **Demonstrate** what the Team has done in the Sprint and get Feedback from the involved.

Sprint Review Meeting

- **The purpose with the Sprint Review is to have a complete review of all the tasks/user stories that should be completed in the Sprint (Sprint Backlog).**
- On the last day of the sprint, your team meets with your product owner, customers, and stakeholders to accept completed work and to identify new requirements.
- **In this meeting, your team demonstrates each task/user story that it completed in the sprint.**
- In many cases, your customers will understand their additional needs more fully after seeing the demonstrations and will identify and discuss the changes that they want to see.
- Based on this meeting, some tasks/user stories will be accepted as complete. Incomplete tasks/user stories will remain in the product backlog, and new tasks/user stories will be added to the product backlog.

Sprint Review Meeting



You   13min

<http://scrumtrainingseries.com/SprintReviewMeeting>

Using DevOps to create/update the Backlog

usn21 / Development / Boards / Backlogs

Search

Development Team

Backlog Analytics

+ New Work Item Product Backlog Item Add to top

Order	Work Item Type	Title	Iteration Path
+ Feature	Web Application	Development\Sprint 1 - Alpha	
Product Backlog Item	> A Web Application should be created	Development\Sprint 1 - Alpha	
Product Backlog Item	> Misc information should be possible to store for each tool	Development\Sprint 1 - Alpha	
Feature	Desktop Application	Development\Sprint 1 - Alpha	
Product Backlog Item	The system should track tools in the inventory, i.e., track when someone borrow tools and whe...	Development\Sprint 1 - Alpha	
Product Backlog Item	The inventory system should have access control	Development\Sprint 1 - Alpha	
Feature	Documentation	Development\Sprint 1 - Alpha	
Product Backlog Item	> A SRD document should be created	Development\Sprint 1 - Alpha	
Product Backlog Item	> A SDP document should be created	Development\Sprint 1 - Alpha	
Feature	Database	Development\Sprint 1 - Alpha	
Product Backlog Item	> The Database should have a proper Table Design	Development\Sprint 1 - Alpha	
Product Backlog Item	> Det system should store data in a Database	Development\Sprint 1 - Alpha	
Feature	Cloud System	Development\Sprint 3 - RC	
Product Backlog Item	> The System should be deployed to Azure	Development\Sprint 1 - Alpha	
Feature	Product Web Page	Development\Sprint 2 - Beta	
Product Backlog Item	> A HTML Web Site should be created	Development\Sprint 1 - Alpha	
Feature	Project Management	Development\Sprint 1 - Alpha	
Product Backlog Item	> Requirements Analysis should be part of the project	Development\Sprint 1 - Alpha	

Backlog items

Planning

Drag and drop work items to include them in a sprint.

Development Team Backlog

Sprint 1 - Alpha **Current** 2021-02-15 - 2021-03-02
Planned Effort: - 12 working days
11 9

Sprint 2 - Beta
No work scheduled yet

Sprint 3 - RC
No work scheduled yet

Sprint 4 - RTM
No work scheduled yet

+ New Sprint



Sprint Planning

Sprint Planning

- Have a Sprint Planning Meeting within the Team
- Move selected Items from the “Product Backlog” to the “Sprint Backlog” (“Beta iteration”)
- Both the “Product Backlog” and the “Sprint Backlog” should be created in Azure DevOps

See Next Slides for more details...

Sprint Planning Meeting



<http://scrumtrainingseries.com/SprintPlanningMeeting>

Purpose with this Meeting:

Planning what to do in the next Sprint (Iteration)

Sprint Planning Meeting



- The purpose with the Sprint Planning is to discuss and select the work items for the next Sprint.
- You select work items from the Product Backlog into the next Sprint Backlog.
- Make sure to set the Deadline for the next Sprint as well.

Sprint Backlog and Taskboard

Azure DevOps usn21 / Development / Boards / Sprints

Search []

February 15 - March 2
7 work days remaining

Sprint 1 - Alpha | Person: All

Development Team

Taskboard | Backlog | Capacity | Analytics | + New Work Item | Column Options

Collapse all

To Do 17 h

- 5 Requirements Analysis should be part of the project
Unassigned | State: New
- 1 A HTML Web Site should be created
Unassigned | 12 h | State: New
- 31 The System should be deployed to Azure
Unassigned | 3 h | State: New
- 13 A Web Application should be created
Unassigned | 5 h | State: New
- 17 Misc information should be possible to store for each tool
Unassigned | 5 h | State: New
- 10 The system should track tools in the inventory i.e., track when someone borrow tools and when

In Progress 8 h

- 11 Create Web Site using the Web Server at USN
Hans-Petter H... | 4 | State: To Do
- 15 Learn basic HTML
Hans-Petter H... | 8 | State: To Do
- 18 Create ASP.NET Core Web Site for Inserting Information for a specific
Hans-Petter H... | 3 | State: In Progress
- 14 Learn basic ASP.NET Core
Hans-Petter H... | 5 | State: In Progress
- 19 Create ASP.NET Core Web Site for Updating Information for a specific
Unassigned | 5 | State: To Do

Done

- 6 Create Initial SRD Document
Hans-Petter H... | State: Done

Project settings <<



Implementation

Implementation

- Start Implementing/Coding
- If you haven't started the coding already, now its the time!!
 - Do work according to the Product/Sprint Backlogs
- The next 4 weeks our main focus will be Implementation/Coding of our System

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

E-mail: hans.p.halvorsen@usn.no

Web: <https://www.halvorsen.blog>

