

Projektityö: "Demo", CSS-Grid, Design Sprint ja Verkkójulkaisemisen työkaluista

CS-C1180 Verkkójulkaisemisen perusteet (5 op)

Pauli Laine, 16.4.2019

”Luennon jälkeen tiedän miten aloitan projektityön tekemisen ja myös mitä työkaluja ja palveluja voin käyttää verkkojulkaisemisessa. Lisäksi tiedän hiukan CSS-Gridistä ja Design Sprintistä”

CSS Grid

CSS Grid

GRID on kenties helpoin tapa hallita elementties sijoittelua CSS:n avulla.

CSS Grid Layout excels at dividing a page into major regions, or defining the relationship in terms of size, position, and layer, between parts of a control built from HTML primitives.

Like tables, grid layout enables an author to align elements into columns and rows. However, many more layouts are either possible or easier with CSS grid than they were with tables. For example, a grid container's child elements could position themselves so they actually overlap and layer, similar to CSS positioned elements.

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout

CSS Grid käyttö

Gridin avulla hallitaan elementtien sijoittelua (suurehkojen) rivien ja sarakkeiden avulla. Esim. sivulla voi olla viisi saraketta ja kolme riviä. Ne on määritelty CSS:n puolella luokkina ja niitä käytetään HTML:n puolelta luokkina.

CSS

```
.wrapper {  
  display: grid;  
  grid-template-columns: repeat(3, 1fr);  
  grid-gap: 10px;  
  grid-auto-rows: minmax(100px, auto);  
}  
.one {  
  grid-column: 1 / 3; //sarakkeilla yhdestä kolmeen  
  grid-row: 1; //rivillä yksi  
}
```

.wrapper

Määrittelee gridin yleiset ominaisuudet

Gridin elementtien välisen tilan ja tyylin

Elementtien leveyden ja sijoittumisen automaattisuuden

HTML

```
<div class="wrapper">  
<div class="one">One</div>  
<div class="two">Two</div>  
<div class="three">Three</div>  
<div class="four">Four</div>  
<div class="five">Five</div>  
<div class="six">Six</div>  
</div>
```

Eli yllä määritellään divit, joille jokaiselle on määritelty CSS:n puolella rivi-sarake positio.

HTML järkevät luokan nimet

```
<div class="wrapper">  
  <div class="ylaosa">Ylä </div>  
  <div class="vasenosa">Vasen</div>  
  <div class="keskiosa">Keski</div>  
  <div class="alaosa">Ala</div>  
  
</div>
```

CSS järkevät luokannimet

```
.wrapper {  
  display: grid;  
  grid-template-columns: repeat(4, 1fr);  
  grid-gap: 10px;  
  grid-auto-rows: minmax(100px, auto);  
}  
.ylaosa {  
  grid-column: 2 / 4;  
  grid-row: 1;  
}  
.vasenosa {  
  grid-column: 1;  
  grid-row: 1 / 7;  
}
```

Vasen

Ylä

Keski

Grid template-columns - FR yksikkö

With CSS Grid Layout, we get a new flexible unit: the Fr unit. Fr is a fractional unit and 1 fr is for 1 part of the available space.

Grid linkkejä

Varsin hyvä alkututoriaali:

<https://medium.freecodecamp.org/learn-css-grid-in-5-minutes-f582e87b1228>

Grid esimerkkejä:

<https://gridbyexample.com/examples/>

Yleisten asettelujen tekeminen gridin avulla:

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Realizing_common_layouts_using_CSS_Grid_Layout

Grid ja Responsiivisuus

https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Grid_Layout/Realizing_common_layouts_using_CSS_Grid_Layout#A_responsive_layout_with_1_to_3_fluid_columns_using_grid-template-areas

<https://medium.com/samsung-internet-dev/common-responsive-layouts-with-css-grid-and-some-without-245a862f48df>

Verkkójulkaisemisen palvelut

Taustaa

Verkkojulkaisemiseen tarvitaan palvelin, jolle sivut tallennetaan.

Staattisten sivujen (HTML,CSS,Javascript) tallentaminen on suhteellisen yksinkertaista.

Mikäli tarvitaan tietojen tallentamista tarvitaan palvelimella tietokanta ja tietokannan ja asiakkaan välinen toiminnallisuus

Tietoturvan ja tietokannan hallinta voi olla vaikeaa

Full Stack-koodaaminen tarkoittaa koko ketjun osaamista asiakkaan näkemistä verkkosivuista palvelimen tietokantaan.

Palvelu vai itsetehdyt sivut

PALVELU

- nopea
- tyylikäs
- varma
- kallis
- helppo
- rajoittava

ITSETEHDYT SIVUT

- vaikea
- halpa, ilmainen
- ylläpito
- turvallisuus
- rajaton

Joitain valmiita palveluita

Wordpress

Wordpress.com

Blogger

Joomla

Drupal

Kotisivukone

Wix

Kotisivukone

Maksullinen (alk n. 4e/kk)

Luvataan helpoksi, tarjoaa valmiita sivupohjia

Vertailulinkkejä

<http://www.wpbeginner.com/opinion/wordpress-vs-blogger-which-one-is-better-pros-and-cons/>

<https://www.joomla.fi/ohjeita>

<https://websitesetup.org/cms-comparison-wordpress-vs-joomla-drupal/>

WordPress.com vs WordPress.org

.org on ilmainen, muokattava. Edellyttää varsin paljon osaamista ja asennustyötä.

.com on osin maksullinen, mm. muokkaus edellyttää maksamista

<http://www.wpbeginner.com/beginners-guide/self-hosted-wordpress-org-vs-free-wordpress-com-infograph/>

Wordpress.org asennus

Edellyttää, että osaat ottaa käyttöön palvelimesi tietokannan.

Edellyttää ymmärrystä palvelimen kansioden ja tiedostojen oikeuksista, poluista yms.

Siirtäminen toiselle palvelimelle esim. koulun koneelta voi olla vaativaa.

Toisaalta wordpressin asennuksen osaaminen on keskeistä verkkoasiantuntijan tietotaitoa.

Kannattaa asentaa koulun koneelle oppimismielessä!

<https://wiki.metropolia.fi/display/tietohallinto/WordPress>

https://codex.wordpress.org/Installing_WordPress#Famous_5-Minute_Installation

Blogger

Googlen omistama, et omista sivustoasi

Valmis alusta, ei tarvitse asentaa tietokantoja tms.

Verkkohotellit

Verkkohotellit tarjoavat maksua vastaan helpotettua palvelua ja tietokantoja yms. Palveluissa on suuria eroja.

<https://web-hosting.thetop10sites.com/>

Oman palvelimen perustamisesta joitain huomioita

Oman palvelimen ylläpito vaatii runsaasti teknistä osaamista.

Asennus

Tietoturva

Varmuuskopiointi, ylläpito

Katso esim. opinnäytetyö palvelimen asentamisesta:

https://www.theseus.fi/bitstream/handle/10024/111749/luostarinen_jani.pdf?sequence=1&isAllowed=y

"Sonera ei kiellä palvelimen pitämistä, mutta eivät myöskään yksityis liittymiin tarjoa kiinteää IP-osoitetta. Yritys liittymiin saisi. Kaikki riippuu tietysti millaista palvelua ja kenelle haluaa tarjota. Itse käytän omille projekteille omaa palvelinta kotona ja nimipalvelut on dy.fi kautta, jota tosin joutuu säännöllisesti päivittämään. Mutta ajastetulla scriptillä tuo menee kivasti.

Vaatii hieman osaamista saada järjestelmä turvalliseksi ja saada se pysymään turvallisena. Saati mitä tehdään kun todetaan että kaikesta huolimatta palvelimesi on hakkeroitu."

Yhteenveto

- Verkkojulkaisemiseen on vaihtoehtoja itsetehtyjen sivustojen lisäksi.
- On olemassa lukuisia palveluita, jotka mahdollistavat kehittyneen verkkojulkaisemisen.
- Jotkin palveluista ovat maksullisia.
- Käyttäjän toiminta on usein rajoitettua (esim. ulkoasun muokkaaminen on maksullista).

Tehtävänänto

Demo

Kurssin projektityönä toteutetaan "sosiaalisen median palvelun toimiva konseptidemo" (lyhyesti: Demo)

Demo voi olla suppeahko yhteisöllisen median palvelun toimiva esittely (esimerkiksi jonkinlainen mini-Twitter/Snaphchat/Facebook). Demoa kokeilevan käyttäjän tulee pystyä käyttämään keskeisiä toimintoja: listata kaverit, lisätä kaverit, lähettää viesti, nähdä saapuneet viestit. Lisäksi voit keksiä itse uusia ja kokeellisia toimintoja.

Mahdollisia toimintoja

Joitain mahdollisia toimintoja:

- Lähetä viesti
- Tykkää viestistä
- Kommentoi viestiä
- Tykättyimmät viestit muuttuvat uutisiksi etusivulla
- Viestien järjestyminen pääikkunassa suosituimmuuden avulla
- Mitä 'ikkunoita' tai kenttiä tarvitset? Kts. esim. Twitter, Instagram tai Facebook.

Tehtävänanto

- Kurssin projektityönä toteutetaan "sosiaalisen median palvelun toimiva konseptidemo" (lyhyesti: Demo) opettujien asiakkaiden web-tekniikoiden avulla (katso *Vaatimukset*). Projektityö tehdään **kahden (2) hengen ryhmissä** ja upotetaan osaksi molempien ryhmän jäsenten verkkosivustoa. Projektityön on liityttävä ainakin toisen ryhmän jäsenen kurssin alussa valitsemaan *teemaan*, muuten aihe on vapaa.
- Demo voi muistuttaa esim. mini-Twitteria/Facebookia/Snapchatia. Uudet toiminnalliset ideat katsotaan eduksi.
- Projektityö palautetaan 15.5.2018 kello 18.

Arvostelu

- 0-4 pistettä *Vaatimukset:* Tehtävänannon mukaisuus.
- 0-4 pistettä *Idea:* Omaperäisyys ja teemaan sopivuus.
- 0-4 pistettä *Käytettävyys:* Toiminnot ja käyttökokemus.
- 0-4 pistettä *Tekninen toteutus:* Ohjelmakoodin laatu ja toimivuus.
- 0-4 pistettä *Grafiikka ja dokumentointi:* Selkeys ja visuaalisuus sekä dokumentaatio (sekä koodissa, että selvitys käyttöönnotosta).

Läpipääsyn edellytyksenä on, että jokaisesta osa-alueesta saa vähintään yhden pisteen.

Palvelun ominaisuuksia?

Yhteisöllisen palvelun prototyyppi/konsepti

16.4.2019

Päivän ohjelma

- Lopputehtävän esittely
- Pulmallisten koodiasioiden ratkaisuja
- Olemassaolevien palveluiden esittely
- Design Sprint -> ideat projektille
- Loppuun varattu aikaa neuvonnalle!

Esimerkkikoodin avaus

-käydään läpi hyvin suppean järjestelmän toiminnallisuus

Osatehtävä olemassaolevien palveluiden analysointi (benchmarkkaus)

Tutki muutamia erilaisia sosiaalisen median palveluita ja mieti mitä toiminnallisuuksia niissä on.

Pohdi kuinka toiminnallisuudet on mahdollisesti toteutettu.

Luonnosasteella olevan palvelun arviointi

Tutki hyvin viimeistelemättömän palvelun toimintaa ja kirjaa ylös toivomasi parannukset ja havaitsemasi virheet.

Viestijärjestelmä

Tiedon tallentaminen

Forum

Palvelun oliorakenne

Missä on viestien lista?

Minkälainen olio on viesti?

Onko olemassa ystävälusta?

Onko jokaisella ystävällä/henkilöllä oma viestilista?

Tietoturvalisuus

Sisääkirjautuminen ja rekisteröityminen

Luottamuspiiri eli ystävät

Kenelle viesti menee?

Näytä minulle tulleet viestit

Laadi funktio, joka näyttää minun lähettämäni viestit tai viestit joiden lähettäjän ystäväpiirissä olen

Tykkäyksen lisäys

Laadi toiminto, jossa voit lisätä tykkäyksen yhteen viestiin

Palvelimen käyttö

Jos et käytä pelkästään local-storagea tiedon tallentamiseen, voit

Local-storagen avulla voidaan kokeilla palvelua yksin tai ns. hot-seat tyyppisesti, eli eri käyttäjä tulee kokeilemaan saman koneen äärelle.

Jos palvelua halutaan testata verkossa tai vaikkapa samassa koneessa useiden istuntojen välisesti, tiedot tulee tallentaa palvelimelle. Palvelin voi olla joko samassa koneessa (local server) tai esim. Aallon opiskelijapalvelimella.

Palvelimessa oleva rajapinta (api)?

Palvelimessa oleva rajapinta

Yksinkertaisin mahdollinen:

Kaksi funktiota: tallennus ja lataus

Tallennus: Local storagesta tallennetaan yksi json-stringi, jonka palvelimen (esim. PHP-skripti) tallentaa palvelimen tekstitiedostoon

Lataus: Palvelimella oleva funktio avaa tiedoston ja palauttaa sen asiakkaalle tekstinä.

Etuja: tiedon muoto voi olla mikä tahansa json-muotoinen merkkijono, kunhan asiakaspään funktiot osaavat tallentaa ja purkaa sen.

Haittoja: hidas (pienikin muutos edellyttää kaiken tiedon pakkaamista/purkamista jne.

Ryhmänmuodostus

Ryhmistä

- Projektityö tehdään pareittain, ja parit muodostetaan Viestintä 2 - kurssin ryhmien sisällä. Neljän hengen ryhmästä tulee siis kaksi paria, viiden hengen ryhmästä yksi kolmen hengen ryhmä ja yksi pari, kolmen hengen ryhmä toteuttaa projektityön yhdessä.
- Muista ilmoittaa projektityön palautuksen yhteydessä parisi nimi!

Palvelusuunnittelusta

Palvelusuunnittelusta

- Ota huomioon mitä tarpeita/haluja käyttäjällä on.
- Ota myös huomioon teknologian rajoitukset (mitä pystyt tekemään/koodaamaan).
- Molempien seikkojen huomioiminen toimii usein hyvin parityössä.

Eräs suunnittelutapa: Design Sprint

- Ongelman ymmärtäminen ja määrittely
- Ideointi
- Ideasta suunnitelmaluonnos
- Prototyyppi
- Proton evaluointi -> palataan tekemään muutoksia (niin alkuun kuin on tarpeen) -> iteroidaan.

Ideointi

- Brainstorming
- Mindmap
- Crazy eight
-

Prototypointi

- Nopeahko kehityskierto
 - Plenin Toimiva Sovellus (Minimum Viable Product) esim. kerran päivässä
 - Inkrementaalinen kehitys: ensin vähän toimintoja, lisätään vähitellen siten että toimivuus säilyy.

Testaus

- Alkuvaiheessa toimintoa voi ja pitää testata itse (nopea testaus->korjaus-sykli)
- Loppuvaiheessa on hyvä testata kehittämisryhmän ulkopuolisilla testihenkilöillä

Ideointiharjoitus

Crazy 8

DESIGN SPRINT

Lisätietoa englanniksi
Pauli Laine
Kevät 2018

What is Design Sprint?

A fast way to design products and services. Originates from Google and Google Ventures. Normally one week, but can be done in few days.

<http://www.servicedesignsprints.com/>

How to find Ideas: Some Ideas

Reliable, non-mocking atmosphere

Explain your idea to someone

Brainstorming

How to find Ideas: Some Ideas contd.

Try not to be too fantastic

But, put too strict technical/legal/business limitations aside for awhile

<https://www.iamnotmypixels.com/how-to-use-crazy-8s-to-generate-design-ideas/>

Design Sprint examples - GV - Google Ventures

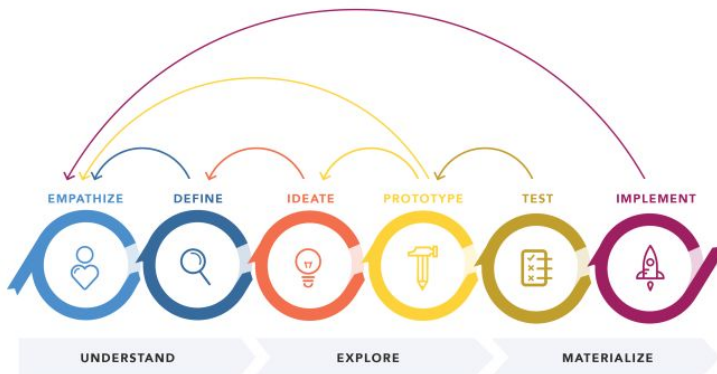
<http://www.gv.com/sprint/>

On [Monday](#), you'll map out the problem and pick an important place to focus. On [Tuesday](#), you'll sketch competing solutions on paper. On [Wednesday](#), you'll make difficult decisions and turn your ideas into a testable hypothesis. On [Thursday](#), you'll hammer out a high-fidelity prototype. And on [Friday](#), you'll test it with real live humans.

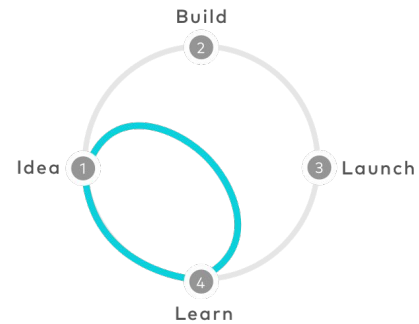
Design Sprint examples - Google

<https://designsprintkit.withgoogle.com/>

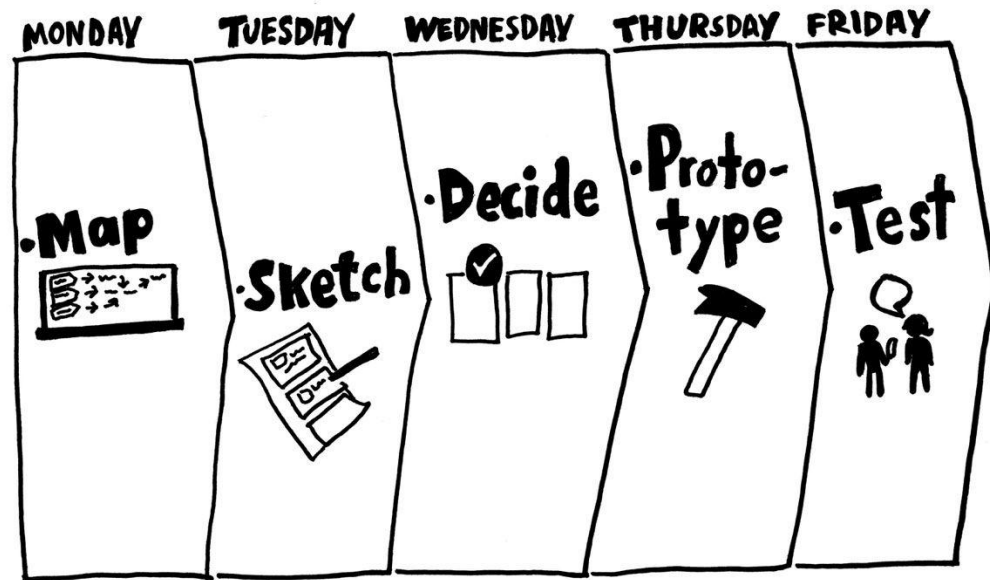
Process of Design Sprint - Some images



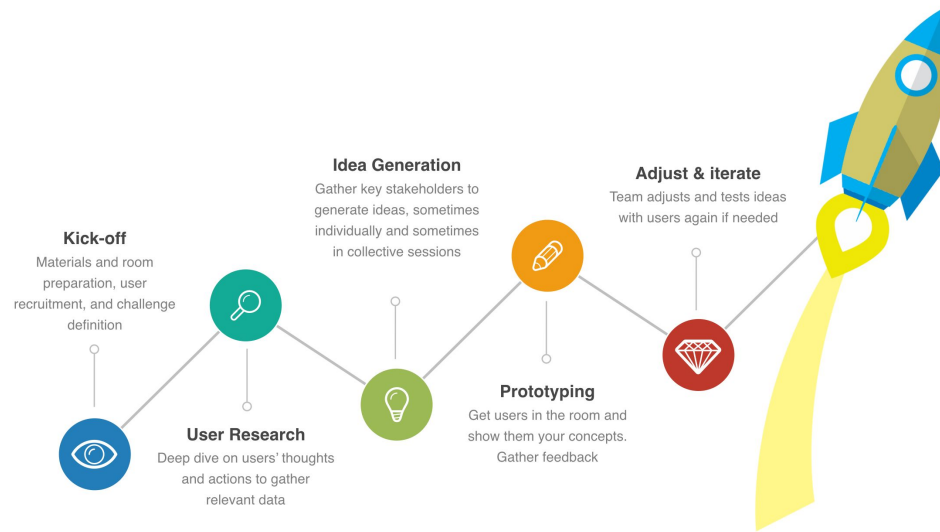
DESIGN THINKING 101 NNGROUP.COM



Process of Design Sprint - Some images



Design Sprint - more pictures



Design Sprints

From Richard Banfield, C. Todd Lombardo, and Trace Wax's the new book, **Design Sprint: A Practical Guidebook for Building Great Digital Products**, here is a breakdown of what a Design Sprint is and why it's important for developing products.

XPLANATIONS™ by STRUCC

How do you do it?



Why is it important?

The focus of a Design Sprint is not to create a finished product, but to speed up learning by rapidly creating and testing rough prototypes.



Move from risk to certainty.

To make productive learning about what works and what doesn't while validating or disproving assumptions, eliminate risks, increase proof chances for success.



Get your unaligned team on board and working together.

Internal conflict and conflict can slow down your team. Sprints build alignment momentum by making everyone



Turn customer confusion into customer delight.

Customers have more choices than ever. If they don't "get it" quickly, the gains is lost. Sprints ensure that you are giving customers something they truly want.

Phases according to Wikipedia

Understand: Discover the business opportunity, the audience, the competition, the value proposition, and define metrics of success.

Diverge: Explore, develop and iterate creative ways of solving the problem, regardless of feasibility.

Converge: Identify ideas that fit the next product cycle and explore them in further detail through storyboarding.

Prototype: Design and prepare prototype(s) that can be tested with people.

Test: Conduct 1:1 user testing with (5-6) people from the product's primary target audience. Ask good questions.

Understanding

Understand the customer and context.

<https://thoughtbot.com/product-design-sprint/guide/understand/problem-statement>

<https://thoughtbot.com/product-design-sprint/guide/understand>

IDEATE - DIVERGE

DIVERGE : illuminating all of the possible paths.

Lot of sketching (pen and paper)

Discussions

Converge

During Converge our goal is to take all of the possibilities that we have exposed over the past two days and hone in on a single version of the prototype that we will build tomorrow.

Throughout Converge, the team should be thinking about the bigger assumptions that are the most important to test. This should lead to a discussion about what type of prototype will do the best job of validating or invalidating it. Activities such as Assumption / Test Table and Final Storyboard should help the team engage this conversation.

From Thoughtbot.com

Prototype

We talk more about prototyping in workshop support slides.

Test

User testing

In fast design sprint you do not need external testers

You, your team, next team etc.

Discussions with the user

User testing metrix

When approaching final phase of product you may want to make proper user testing and collect data.

WORKSHOP

Support slides

Our workshop is a modified version of Google Design Sprint

<https://designsprintkit.withgoogle.com/resources/>

PHASES:

Find and define a problem

Ideate around the problem - diverge

Sketch a solution - Converge

Make a prototype - test and present it

Evaluate and iterate

Design Sprint - Finding and defining a problem

"The point of Day 1 is to define the problem, and a critical piece of this is being **able to clearly outline the user, technical, and business requirements**. Without these, a team can't complete Day 1 of the sprint in any meaningful way."

"The prep included collecting and analyzing all the user roles - their goals, needs, and pain points. In addition to Rachel's pre-sprint interviews, information was gathered via diary studies, task analysis, UX research and team surveys. The important piece was to document not just what people did in their separate roles, but the reasoning behind each decision. "

Who is customer in our workshop?

Discussion and solutions

Workshop session - Find and define a problem

Some suggestions:

From your life

Not too difficult, simple

Easy to talk about

10 Minutes, write your ideas

10 minutes coaching session

10 minutes put the ideas to a poster, discussion

Understanding and ideate

Write down current understanding a problem. Then we go to ideate-phase.

Exercise - Brainstorming - Mind Map

Brainstorming

- Team ideates

- No criticism, no laughing (at ideas, otherwise yes)

Mind map

- Draw linked ideas to paper

- Start from some concept

Exercise - Brainstorming - Mind Map

Mind map

Online tools and tool reviews:

<https://coggle.it/gallery>

[https://mashable.com/2013/09/25/mind-mapping-tools/
#_allsvzQt8qP](https://mashable.com/2013/09/25/mind-mapping-tools/#_allsvzQt8qP)

<https://bubbl.us/07291151942481142>

Exercise - Coaching

Idea owner explains the idea to a neutral listener (coach)

Coach asks for clarifications for anything unclear

Explaining idea really helps to formulate it better

Exercises from google - Crazy 8

<https://designsprintkit.withgoogle.com/methods/sketch/crazy-8s/>

Solution Sketch

<https://designsprintkit.withgoogle.com/methods/sketch/solution-sketch/>

Decide and converge

<https://designsprintkit.withgoogle.com/methods/decide/>

Prototyping

Paper prototyping

Prototyping tools

Javascript

Prototyping with Javascript

Minimal page

Minimal amount of code

Database and data can be simulated without the server

-huge savings in coding time

Typical time from idea to initial prototype: 30min or less

-each iteration after that just few minutes

Learnings from AGILE Development

Stand in meeting

Minimum viable product (MVP)

Incremental design

Continuous interaction between designer and developer

Exercise - Prototype

Paper proto

Fast, easy to make

Requires human interaction

Prototyping tools

<https://www.creativebloq.com/web-design/top-10-prototyping-tools-2016-21619216>

Coding prototypes

If you need real interaction (for example with real computation)

Paper prototyping

Fast, reliable, cheap
Requires facilitator
Some techniques:



New paper to represent new page

Holes in papers

Additional small paper clips

<https://www.uxpin.com/studio/blog/paper-prototyping-the-practical-beginners-guide/>

Coding prototypes -using HTML+CSS+Javascript

You can show REAL functionality

Time, data, user interaction etc.

Can be as fast as prototyping tool

Can be automated

Automated calculation of time spent

clicking etc

Current HTML/CSS/JS very good!

Coding can be slow

Requires skills

Visuals may be not so cool

LINKS and READING MATERIAL

<http://www.gv.com/sprint/>

<https://designsprintkit.withgoogle.com/>

<https://medium.com/project-management-learnings/design-sprints-at-google-85ff62fed5f8>

<https://www.iamnotmypixels.com/how-i-use-the-design-sprint-process-for-different-types-of-projects/>

Evaluate and critic

When to discuss the problems?

When to change the design?

Develop-test-evaluate-develop loop

Many iterations normally needed (tens, hundreds...)

So the faster the loop the better

Keyboard shortcuts, several windows open

Working in very small groups/pairs

Designer/Developer can do self-testing self-evaluation

Real user testing is not needed and it is not possible
during every loop iteration.



KIITOS!