

# VERKTØY

---

## Innhold

Verktøy .....	1
1. Scrumwise.....	1
2. Trello.....	1
3. Github .....	1
4. Eclipse.....	2
5. PostgreSQL.....	2
6. Dropbox .....	2
7. Google Maps API .....	3
8. Office Word .....	3
9. Office Excel og OmniGraffle Pro.....	3
10. Photoshop, Illustrator.....	3
11. Facebook.....	3
12. Nettbrett og android telefon til testing.....	4

## 1. SCRUMWISE

---

En av fordelene ved scrum er at hele teamet skal kunne observere og ha oversikt over hva som skjer i prosjektet til enhver tid. Det lot seg ikke gjøre å ha stand up møte hver dag, ettersom alle har forskjellige timeplaner og ikke har tid til å møtes hver dag. Vi mente derfor at vi trengte et verktøy til å holde oversikt over hvem som jobbet på de ulike oppgavene og hvordan vi lå an med sprinten fortløpende. Allerede første møte bestemte vi oss for å prøve Scrumwise som er et verktøy der all prosessinformasjon ligger lagret på et sted. Den gir god oversikt over både backlog, sprinter, taskboard, estimerer og genererer et burndownchart. Vi valgte på slutten av første sprint å slutte å bruke Scrumwise, fordi det ble lite brukt og utover en måneds prøveperiode ville vi måtte betale. Vi slo fast at Scrumwise var unødvendig ettersom at vi brukte det så lite, og vi ønsket å bruke mer tid på å lære oss Android programmering enn å sette oss inn i Scrumwise.

## 2. TRELLO

---

For å erstatte noe av Scrumwise-funksjonene begynte vi å bruke Trello for å få oversikt over oppgaveprogresjon. Trello er et web-basert samarbeidsverktøy som organiserer prosjekter inn i "boards", eller tavler. Vi laget et nytt board for hver sprint, og la oppgaver knyttet til sprintene inn i disse. På denne måten har vi hatt oversikt over hvilke oppgaver som var påbegynt, hvilke som var ferdige og hvilke vi ikke hadde startet på. I tillegg har vi kunnet se hvem som jobbet med hva, legge til kommentarer til oppgavene og vise eventuelle oppgaver som måtte flyttes til neste sprint. Flere på gruppa hadde erfaring med bruk av trello fra før og det har derfor blitt mer brukt, i tillegg til at vi mener Trello har vært mye mer oversiktlig enn Scrumwise.

## 3. GITHUB

---

Github er hovedsakelig et web-basert versjonskontrollsystem. I den typen utviklingsprosess som vi har vært gjennom, der flere personer koder på samme kodebase samtidig, er et versjonskontrollsystem veldig viktig. Det systemet passer på er at ingen kode går tapt, man vet hvem som har gjort hvilke endringer, og den prøver å forhindre konflikter hvis to personer har jobbet på samme fil.

Måten man jobber med Github på er at hovedkodebasen ligger i skyen på Github sine servere, og når man endrer på kodebasen lager man en forgreining av denne. Dette betyr at du kopierer kodebasen ned til din lokale datamaskin, for så å gjøre endringer. Når man har foretatt seg de endringene man skulle, må man dytte koden opp til Github igjen. Når dette gjøres slår man sammen endringene med den originale koden. Github vil så prøve å flette dette sammen automatisk, noe som vanligvis går veldig bra, men noen ganger klarer den det ikke. Da må man selv ta stilling til hvilke endringer/versjoner av koden som skal være med i den nyeste versjonen.

Github har flere fine funksjoner som et hvert versjonskontrollsystem burde ha; den lagrer alle endringer siden du startet prosjektet, og du kan derfor rulle tilbake til en tidligere versjon av koden om noe skulle gå galt. I tillegg til å være et versjonskontrollsystem tilbyr Github en del andre funksjonaliteter. Du har muligheter for å legge til såkalte "issues" som man delegerer til de

som jobber på prosjektet. På denne måten kan du se hva hver person skal gjøre. Hver eneste kodelinje i kodebasen kan spores til en person, og man vet da hvem som har skrevet hva.

Vi fikk tilbud om en SVN server fra studieassistenten, men siden noen på gruppa hadde erfaring med Github fra før bestemte vi oss for å bruke dette. Det skal også nevnes at det er gratis.

#### 4. ECLIPSE

---

Da vi bestemte oss for å gå for Androidutvikling ble vi rådet til å laste ned ADT Bundle, ettersom det egner seg veldig bra for uerfarne Androidutviklere. ADT Bundle inkluderer viktige Android SDK komponenter og en versjon av Eclipse(et kodebehandlingsprogram) med innebygd ADT(Android Developer Tools). Fordelen ved bruk av Eclipse fra ADT Bundle var at det var raskt og enkelt å komme i gang. De innebygde ADT funksjonene gjorde det lett for oss å opprette et nytt Android prosjekt, lage UI til applikasjonen og debugge applikasjonen ved hjelp av Android SDK-verktøyene. Alt koderelatert var i tillegg lett tilgjengelig innenfor programmet, noe som gjorde det veldig oversiktlig å jobbe med.

#### 5. POSTGRESQL

---

For å lagre data fra flere bønder var i nødt til å ha en database. Valget vårt havnet raskt på PostgreSQL. Årsaken til det, var at den eneste i gruppa som kunne litt om databaser fra før, kjente til dette databasehåndteringssystemet. I tillegg har PostgreSQL åpen kildekode som er veldig solid og stabil, og det er ingen restriksjoner ved bruk av den. Vi har valgt å ta backup av all dataen i databasen hver natt slik at hvis en systemkrasj skulle inntreffe vil vi ha all dataen lagret på en annen server. Databasen er ikke tilgjengelig utenfra. Dette vil si at man i vårt system må gå gjennom Raspberry PI for å få kontakt med den, noe som andre databaser ikke nødvendigvis ville krevd. Ellers ville vi ikke hatt noen mulighet til å sanitere("rengjøre") SQL spørringene som kommer inn. Dette er viktig siden systemet ellers ville vært svært åpne for SQL-injeksjon – altså å bli angrepet – og resultere i at den som angriper får tilgang på sensitiv brukerinformasjon. Vi har derfor valgt å ha et mellomlag mellom database og applikasjon som saniterer data og håndterer eventuelle feil.

#### 6. DROPBOX

---

For at alle på gruppa skulle få tilgang til alt vi produserer av diagrammer, dokumenter, bilder, osv., opprettet vi en felles Dropbox-mappe. Dropbox er et delingsverktøy der man lagrer filer i mapper som man har mulighet til å dele med andre, noe som gir god støtte for samarbeid.

Når vi har arbeidet med dokumentasjonen underveis, oppdaget vi en ulempe med bruk av Dropbox. Det var umulig for flere å redigere på samme fil samtidig, uten å overskrive det den siste personen hadde gjort. Dette kunne vi løst bedre ved å bruke Google Drive, men vi valgte å opprette hver sine filer, skrive ferdig for så å få dokumentasjonsansvarlig til å samle alt i et dokument. Denne metoden ble mye brukt under utviklingen av sluttrapporten. Ut over dette er Dropbox et veldig praktisk verktøy for vår bruk, som er lett å håndtere og forstå – som igjen var mye av grunnen til at vi valgte å bruke det som delingsverktøy.

## 7. GOOGLE MAPS API

---

En av hovedgrunnene til at vi valgte Googles karttjeneste, var at å implementere tjenesten på Android er ganske enkelt og sømløst, i forhold til alternativene. Vi så også på muligheten for å erstatte kartene fra Google med kartverkets kart, og da samtidig beholde kartfunksjonene i Google API. Tanken bak var at vi ønsket å sikre et mer detaljert kart, men etter å ha vært i dialog med kunden valgte vi å prioritere andre funksjonaliteter siden kartet til Google fungerte, og det gir mulighet for å se sauene i en type topografisk kart.

## 8. OFFICE WORD

---

Til å håndtere tekst, tabeller og bilder i rapporten, og til møtereferater, har vi stort sett brukt Microsoft Office Word. Dette har ikke vært helt optimalt for de på gruppa som ikke har hatt Office tilgjengelig, da formateringene ødela visningen ved bruk av andre verktøy. Alle har fortsatt kunnet lese og behandle teksten, men det kunne kanskje vært mer aktuelt å benytte LaTeX. Vi valgte å bruke Word ettersom det gjorde det lett for oss å opprette, redigere og formatere både tekst, bilder og tabeller med et visuelt tilfredsstillende resultat.

## 9. OFFICE EXCEL OG OMNIGRAFFLE PRO

---

I tillegg til Word har vi brukt Office Excel og OmniGraffle Pro til å lage tabeller og diagrammer. Vi valgte OmniGraffle som er et gratis UML-behandlingsprogram fordi det gjorde det lettere og raskere for oss å lage UML-diagrammer. Excel ble brukt for å lage og bruke en burndown-chart.

## 10. PHOTOSHOP, ILLUSTRATOR

---

For å lage ikoner, knapper og bakgrunner til applikasjonens grafiske brukergrensesnitt har vi brukt programmene Photoshop og Illustrator fra "Adobe CS6 Master Collection". Illustrator er et redigeringsverktøy for vektorgrafikk og Photoshop er et bilderedigeringsprogram. Grunnen til at vi brukte disse programmene til design, er at gruppas designansvarlig har god erfaring med dem, og at designet skulle bli så bra som mulig.

## 11. FACEBOOK

---

Mye av kommunikasjonen i gruppa, med unntak av under møter og arbeidsøkter, har foregått på Facebook. Facebook er noe vi alle sjekker jevnlig, og det ble derfor naturlig for oss å bruke dette. Vi opprettet en egen gruppe for å holde oversikt på kommunikasjon, avtale møter, gi hverandre beskjeder og spørre om ting vi lurte på.

## 12.NETTBRETT OG ANDROID TELEFON TIL TESTING

---

Vi syntes Android-emulatorene som er innebygd i Eclipse ATD var veldig treg i oppstart og under kjøring. Derfor har vi brukt telefoner og nettbrett med Android til å teste applikasjonsfunksjonene underveis. Scrum-master gikk, på eget initiativ, til innkjøp av et Android-nettbrett som har blitt benyttet som en test- og debuggingsenhet. Enheten er av typen Samsung Galaxy Tab 2(GT-P3110), og det er denne enheten vi har lagt hovedfokuset vårt på under utviklingen, ettersom kunden forsikret oss om at applikasjonen kun trengte å virke på en enhet. Det gis derfor ingen garanti for at applikasjonen fungerer optimalt på andre enheter enn denne.