

Week 4: Report Information

- Sitemaps/ app layouts/ screenflows
- USE Cases - get them right avoid concurrent updates example
- Diagrams: Class/Sequence/ERD/USE level 0 - level 1 diagrams
- Spectrum of skillsets minimal & hopeful/ prgression
- Terminology - avoid jargon
- Identify unique users (carer, client, event poster)
- github use
- prototypes - paper/ wireframe/ screen prototyping
- Law of demeter
- MVC / DAO Guidelines if necessary
- Compile a broad amount of research - (api's available to us, whats been done before?)
- software metrics (Mccabes Cyclomatic complexity)
- Nielsons Heuristitcs
- Tester must: identify testing framework - how to trap errors (try catch) error handling
- TDD - Test driven development, develop incrementally, start small
- Automated/ manual testing
- Observational studies (do an action, think aloud, though processes)
- Note UI Guidelines wealth of data from Apple/ Google documentation
- Agile development path
- Test VS Evaluation
 - Test : Internal - personal within group... binary does it work 0/1
 - Evaluation: External - presented to user... how well does it work 0-10
 - ◆ create personas - characters
 - ◆ present to clients, get past politeness of just saying "yes that would be great" - establish is it good or bad and if so how good or bad? could use checksheets and grade each implementation of each feature.