

## Dashboards

### Introduction

?

### Project Outline

The project will be covered in 3 sections. These are listed below.

**The research/discovery phase** is where you immerse yourself in the project to get the background you'll need to make design decisions later in the project. During this phase you will try to learn as much about your client's business, objectives, users and competitors as possible.

**The design phase** is where you work out how what you are designing will work and how it will fit together. This phase will define its scope, its features and functionality and how it behaves.

**The validation phase** is where you identify whether what you came up with in the design phase actually works with its intended audience. This phase is typically followed by further rounds of design and testing to solve the problems you inevitably find when you test with users.

## UX Brief (Mike, Simon and Zulf)

- **Project summary**
  - Describe what the project is in one paragraph.
  - At a high-level, what is the project, and why is the project is being done?
  - When it's expected to be completed.
- **Business goals**
  - The 'why we're doing this' question?
  - This should align with the business strategy.
  - This question helps UX align to business strategy and can help make decisions about functionality or features later in the project.
- **User goals**
  - What are the main things the user is trying to achieve by using this application?
  - High-level but should be the user's goals, not business goals.
  - What does the user want to get out of interacting with this application?
  - These goals should be referred to as much as possible during the design phase to keep the user the focus in design.
- **Competitors**
  - Who are the primary business competitors?
  - A competitive review should definitely be one of the next steps in the project so getting this info up-front is key.
- **Competitive analysis**
  - How does the business want this application to be better than what the competitors are offering?
  - There should be some business strategy behind the project.
- **Success measurements**
  - This seems so obvious but it is so often not asked, and when asked it is often really hard to get an answer.
  - There are many ways to measure success. Analytics and usability testing are great ways to get some measurement on success, but they're not always an option. Know this up front and find out from the client what will make this project successful.
- **Usability issues**
  - Sometimes this has to be its own meeting, or part of requirements gathering or a heuristic review.
- **Technical constraints**
  - Take some time to find out what existing business rules or technical constraints might rein-in some great concepts that just can't be developed. This could include lack of technical resources, or maintaining code in an older platform, or aligning to an existing database structure.
- **Scope outline**
  - It's useful to have a high-level bullet-list of the project scope. Add priority of items here, so you can discuss this as a team.
- **Out of scope**
  - Specify the things you understand to be out of scope.

## Project plan

- Research and document review (Kick-off meeting)
  - Gathering the Requirements
    - Brainstorming ideas on what to have in the dashboard
      - Concise and clear gadgets/widgets
      - Fixed and movable (position and order)
      - No visible boundaries, subtle/soft look (optional templates)
      - Responsive design with scalable widgets/gadgets
      - Minimalistic and simple, but still aesthetically pleasing to the eye
      - No scrolling (definitely not horizontal scroll bars) on gadget/widget, auto fit size (unless data widget/gadget)
      - Customisable gadgets/widgets based on theme
      - Real time information, updates dynamically
      - Click on gadget/widget to drill down deeper into details
      - Have more than one dashboard available to user
      - Templates – users chooses the template of the gadget
      - Interactive gadget – gadgets that are linked, so when you change the details (filter) of one you effect all the gadgets
      - Expandable gadgets, dynamic gadgets that open on the dashboard when they are clicked
      - Gadgets should hold summary not details
      - Each gadget needs to understand it's rendering capability
      - On mobile view only show certain gadgets, user chooses
      - Widgets/gadgets should be:
        - Metric (text/number),
        - mini graph (pie/bar/line),
        - icons (red/green/amber),
        - images (visual style to gadget/widget),
        - List (data values),
        - Links to other pages,
        - Trend (x and y axis),
        - Status (high/low),
        - comparative data (compares one data to another)
        - web pages
        - noticeboard (template)
        - Query
        - html
        - calendar
        - tabular list
        - gadget/widget template designs (choose template then the data)
        - Allow colspan and rowspan
        - Data should be clear from distraction
        - Visually aligned blocks
        - No borders/dividers (optional)
        - Movable (re-order the gadget/widget)
        - Minimise the gadget/widget
        - Settings option on the gadget/widget
        - When creating gadget have the option of the view is only available on tablet and desktop

- Each gadget has a priority list, so you choose what the priority is on a mobile phone
    - List the current dashboards features and which features are staying and which are going
      - Staying
        - Query (in bridgeit at moment, not fully)
        - Links (in bridgeit at moment)
        - Content
        - Web page
        - Count query
        - Trend
        - Calendar
      - Going
        - Right click remove
        - Remove dashboard option from gadget make it higher
    - Shared understanding of the problems the project aims to solve
      - Current dashboard design is not responsive
      - It is very rigid and not attractive
      - Difficult in presenting data in a small concise way
      - Not re-implementing like-for-like existing products, creating something new
    - What are our objectives in designing the dashboards
      - Move to responsive
      - Make it easier to use
      - Make it visually appealing
      - Make it useful with the data
      - Make updating/deleting/adding very easy
    - What is the vision for the project
      - Mike to answer!
    - Are their Design Patterns in dashboards (Are there common features that all dashboards have)
      - Grid layout
      - Rows and columns
    - Statistics from our current product that can highlight areas of concern
      - Lara and Andy to answer!
    - Deep understanding of users wants and needs for dashboards
      - Lara and Andy to answer!
    - Sitemaps
      - Create dashboard
        - Web Access Visio diagram
        - Console Visio diagram
      - Create widget
        - Technical constraints that may need to be considered
        - Visual constraints that may need to be considered
  - Business Requirements (**UX Brief**)
  - User Tests (what test are we writing, how are we going to perform them etc)
- Identify users (user research)
  - Personas (Short, vivid descriptions of fictional characters who represent a product's users)
    - Support Desk Analysts – (view, drill into data, not add gadgets)
    - System administrator – (Full, creates dashboards, not user of dashboards)
    - End User – (view only)

- Support desk team lead – (view, drill into data, add options available, several dashboards to monitor team stats, unique dashboards)
    - Business Managers – (view, report only purpose, possible drill into data)
      - View high level details on stats on how the team is doing
      - Stats on charges to business
- Requirements
  - Existing Product assessment
    - Customer experience maps (A visualization of a process that users follow before, during and after using a product or service)
      - Is it clear how to progress through the dashboard to add, delete, update, etc.
      - Is it clear what to do next
      - Primary user journeys
      - Journey to add/delete/update the widget/gadget
      - User flows
  - New product requirements
    - Functionality
      - What features are going to be added to the dashboard
      - Types of widgets/gadgets
      - How are they going to work
      - How do you add a gadget
      - How do you delete a gadget
      - Features to add to dashboards
      - Grouping
    - Features
      - Look and feel (header, footer navigation, etc)
      - Responsive (what is the solution going to be)
      - Determine product definition
      - Define architectural design
      - Conduct a content/function/feature inventory list
    - Task Modelling
      - Process flow that matches expected steps and content needs (activities users perform in order to reach their goals.)
    - Customer experience maps (A visualization of a process that users follow before, during and after using a product or service.)
      - Is it clear how to progress through the dashboard to add, delete, update, etc.
      - Is it clear what to do next
      - Primary user journeys
      - Journey to add/delete/update the widget/gadget
      - User flows
    - User journeys (Identify how users flow through your product. Design the structure of your product to ensure users can flow through it efficiently.)
      -
- Strategy workshop (the vision)
- Information architecture (The process of organizing information to make its retrieval as simple as possible.)
- Ideation workshop
  - Collaborative design method to decide which design solution are suitable for the project based on the above
- Sketching
- Wire-framing the journey
- Interactions

- Error notifications
- Success notifications
- Link behaviour
- Drag and drop
- Interactive behaviour
  - Menu/icons/interactions with gadgets that make it clear on how to use them
  - Movable dashboards/grid framework/scalable?
- Prototypes
- High resolution designs
- Interactive designs
- Navigation
- Concept designs
  - Sketches
  - Low-fidelity diagrams
  - High-fidelity diagrams
  - Prototypes
- Usability testing (throughout the project)
  - Robust, flexible, elegant and easy to use
  - User feedback
  - Quality assurance testing

Additional Information yet to be sorted:

- Should you allow users to create their own dashboards or modify published dashboards
- Depending on role you should have the option to create your own dashboard
- System administrator creates a list of gadgets and the user can only choose from the pre-defined list
- Progress status on incidents and requests (1-5, where the highlighted number/block indicates where you are in your process)
-

## Dashboards

- **Gathering the Requirements (1 week including documenting this)**
  - **Brainstorming ideas on what to have in the dashboard**
    - Concise and clear gadgets/widgets
    - Fixed and movable (position and order)
    - No visible boundaries, subtle/soft look (optional templates)
    - Responsive design with scalable widgets/gadgets
    - Minimalistic and simple, but still aesthetically pleasing to the eye
    - No scrolling (definitely not horizontal scroll bars) on gadget/widget, auto fit size (unless data widget/gadget)
    - Customisable gadgets/widgets based on theme
    - Real time information, updates dynamically
    - Click on gadget/widget to drill down deeper into details
    - Widgets/gadgets should be:
      - Metric (text/number),
      - mini graph (pie/bar/line),
      - icons (red/green/amber),
      - images (visual style to gadget/widget),
      - List (data values),
      - Links to other pages,
      - Trend (x and y axis),
      - Status (high/low),
      - comparative data (compares one data to another)
      - web pages
      - noticeboard (template)
      - Query
      - html
      - calendar
      - tabular list
      - gadget/widget template designs (choose template then the data)
      - Allow colspan and rowspan
      - Data should be clear from distraction
      - Visually aligned blocks
      - No borders/dividers (optional)
      - Movable (re-order the gadget/widget)
      - Minimise the gadget/widget
      - Settings option on the gadget/widget
      - When creating gadget have the option of the view is only available on tablet and desktop
      - Each gadget has a priority list, so you choose what the priority is on a mobile phone
  - **List the current dashboards features and which features are staying and which are going**
  - **Shared understanding of the problems the project aims to solve**
    - Current dashboard design is not responsive
    - It is very rigid and not attractive
    - Difficult in presenting data in a small concise way

- **Not re-implementing like-for-like existing products, creating something new**
  - **What are our objectives in designing the dashboards**
    - Move to responsive
    - Make it easier to use
    - Make it visually appealing
    - Make it useful with the data
    - Make updating/deleting/adding very easy
  - **What is the vision for the project (what features will this dashboard have)**
    - (Mike)
  - **Are their Design Patterns in dashboards (Are there common features that all dashboards have)**
    - Grid layout
    - Rows and columns
    -
  - **Statistics from our current product that can highlight areas of concern**
    - Lara and Andy
  - **Deep understanding of users wants and needs for dashboards**
    - Lara and Andy
  - **Sitemaps**
    -
  - **Any good features that dashboards have that we want to consider**
    -
  - **Technical constraints that may need to be considered**
    -
  - **Visual constraints that may need to be considered**
    -
- **Functionality (3/4 days including documenting this)**
  - What features are going to be added to the dashboard
  - Types of widgets/gadgets
  - How are they going to work
  - How do you add a gadget
  - How do you delete a gadget
  - Features to add to dashboards
  - Grouping
  - Menu/icons/interactions with gadgets that make it clear on how to use them
  - Movable dashboards/grid framework/scalable?
- **Features (1/2 days)**
  - Look and feel (header, footer navigation, etc)
  - Responsive (what is the solution going to be)
  - Determine product definition
  - Define architectural design
  - Conduct a content/function/feature inventory list
- **Task Modelling**
  - Process flow that matches expected steps and content needs
- **Customer experience journey (mapping out) (2/4 days including documentation + mapping)**
  - Is it clear how to progress through the dashboard to add, delete, update, etc.
  - Is it clear what to do next
  - Primary user journeys
  - Journey to add/delete/update the widget/gadget
  - User flows

- Usability testing (throughout the project)
  - Robust, flexible, elegant and easy to use
  - User feedback
  - Quality assurance testing
- Ideation workshop (2-6 days depending on how productive the days are)
  - Collaborative design method to decide which design solution are suitable for the project based on the above
- Wire-framing the journey (2-6 days depending on just pages or actual journeys)
- High resolution designs (2-4 days depending on how many pages and how much detail is required)
- Interactive designs (2-4 days)

- Should you allow users to create their own dashboards or modify published dashboards
- Depending on role you should have the option to create your own dashboard
- System administrator creates a list of gadgets and the user can only choose from the pre-defined list
- Progress status on incidents and requests (1-5, where the highlighted number/block indicates where you are in your process)
-