Table of Contents

[Development of Incorrect User Interface 2](#_Toc3460306)

[Introduction 2](#_Toc3460307)

[Strategies 2](#_Toc3460308)

[Surveying and End User Analytics 2](#_Toc3460309)

[Development and Testing 2](#_Toc3460310)

[Review and Implementation 2](#_Toc3460311)

[Recommendations 3](#_Toc3460312)

[Gold Plating 4](#_Toc3460313)

[Introduction 4](#_Toc3460314)

[Strategies 4](#_Toc3460315)

[Recommendations 4](#_Toc3460316)

[Changes to Requirements 5](#_Toc3460317)

[Introduction 5](#_Toc3460318)

[Strategies 5](#_Toc3460319)

[Recommendations 5](#_Toc3460320)

# Development of Incorrect User Interface

## Introduction

Incorrect development of the product UI has been identified as a critical risk of high probability within the risk assessment matrix. Incompatible or incorrect UI elements will eventually lead to user dissatisfaction and ease of use issues that will arise with the implementation of superfluous functionality. Further issues will also arise when implementing training programs for staff to use the software. Additional unnecessary functionality, while often useful for power users, will not need to be implemented within the software for end users, as it exacerbates confusion, and therefor training time, with little benefit to day to operative requirements when using the software.

## Strategies

There are several strategies when dealing with the risk of developing wrong or unnecessary UI’s. These strategies can be divided into three main groups: Surveying and End User Analytics, Development and Testing and Review and Implementation.

### Surveying and End User Analytics

* Survey end users to define the functionality and level of ease of use that will be required by end users. This should be undertaken as an initial step, before development of the software begins.
* Analyse end users of the software, with emphasis on examining mental models in relationship to the user interface.
* Define end user work scenarios in relation to the software and analyse them in terms of interface goals.
* Define user interface goals and requirements based upon the information that surveys and end user analysis have provided. Ensure that the goals and requirements are properly documented and updated when necessary.

### Development and Testing

* Develop a prototype of the user interface in accordance with interface goals and requirements identified at the survey and analytical stage.
* Test the prototype within a sample user group.
* Collect information and feedback from the sample user group.

### Review and Implementation

* Review the data and feedback collected from the development and testing stage and cross reference it with the goals and requirements defined within the surveying and analytics stages.
* Ensure that any new goals or requirements that have arisen from testing are updated and documented.
* Implement changes as requested or necessary to the prototype.
* Return to the testing phase for any changes or new functionality implemented as appropriate.

## Recommendations

* Survey and test the prototype on staff that have been identified as key operatives of the software, and on staff that have reported high levels of discomfort and insecurity on the technology readiness index questionnaire.
* Create and implement a plan for the UI development process using the strategies highlighted above.
* Ensure that the UI designer, and architecture developer, and back end developers are working collaboratively to ensure the UI has appropriate functionality for all end users.
* Develop secondary UI for high level users with further understanding and a broader scope of required functionality. This UI should be aimed towards administrative users.

# Gold Plating

## Introduction

Gold plating has been rated as a relatively critical risk of decent probability within the risk assessment matrix. Gold plating within projects further compounds issues relating to development, often increasing development times and costs, interfering with proper operation and requirements of the software, and in the improper use of resources during development of the product. These issues make Gold Plating a key area of concern, especially in relation to lower budget/lower turnaround projects where gold plating can cut revenue from the product sales by implementing functionality for free.

## Strategies

* Create a development plan that addresses only the functionality defined within the scope of works.
* For larger projects that require multiple teams, delegate tasks appropriately to each team to ensure that resources are used effectively.
* Employ project management techniques and processes that ensure that software development staff are kept on task.
* Allocate additional resources, or resources that free up onto critical areas of functionality defined within the original scope of work that may be behind schedule.

## Recommendations

* Ensure that the scope of works is clearly communicated to staff, and emphasise that work is to be aimed at delivering on the scope, not on extras.
* Implement agile software development practises. Short sprints of work ensure that there isn’t excess time spent in development that is then wasted on gold plating. Additionally, the customer then reviews and accepts the results quickly, ensuring that the product is up to the expected standard of the customer, not the standards of any project team members that may have perfectionist qualities.

# Changes to Requirements

## Introduction

The risk assessment matrix identifies additions or changes to requirements of the project as the most likely and most critical issue to arise during the development process. Changes to requirements that have not been defined within the scope causes several problems during development. It increases the amount of work needed to be completed before any set deadlines. It adds additional features that have not been analysed during costing reviews and analysis during the planning stages of the project, and often it will also interfere with existing functionality, and require additional resources to implement, all at the cost of the developer, unless otherwise agreed upon.

## Strategies

* Define and baseline the requirements for the project prior to commencement.
* Ensure that any additional features or requirements defined by the customer incur additional cost after the scope has been initially defined.
* Ensure that the development team have a solid uniform understanding of what the customers requires with the system.
* Identify existing products on the market with similar functionality that can be used to baseline or establish firm requirements.
* Administer surveys to stakeholders to identity desired user functionality within the system.
* Identify any known change requests that are pending.
* Identify whether the customer has a history of requesting changes beyond the agreed scope.
* Analyse whether the change will impact the performance or the existing desired functionality of the system.
* Identify the resources required to implement the change.

## Recommendations

* Ensure that changes to the scope are identified within the scope of works, and will require additional cost on behalf of the customer to implement changes beyond the originally agreed upon scope of works
* Communicate any requested changes to key stakeholders to ensure that the changes to functionality are necessary.
* Research and analyse existing products already on the market when defining the scope and or changes to the scope.