Contents

Part I
Agile Software Development

1 *Agile Outline*

1. Customer Satisfaction by early and continuous delivery of valuable software.
2. Welcome changing requirements, even in late development.
3. Deliver working software frequently.
4. Close, daily cooperation between business people and developers.
5. Projects are built around motivated individuals who should be trusted.
6. Face to face conversation is the best form of communication.
7. Working software is the primary measure of progress.
8. Sustainable development, able to maintain a constant pace.
9. Continuous attention to technical excellence and good design.
10. Simplicity - the art of maximising the amount of work not done is essential
11. Best architectures and designsemerge from self organising teams.
12. The team regularly reflects on how to become more effective and adjusts accordingly.

1.1 *How can one enforce Agile Software Development?*

- Break product development into small increments that minimise the amount of up front planning and designs.
- Enforce iterations or sprints, which involves a multi-disciplinary team working on: planning, analysis, design, coding, unit testing, and acceptance testing.
- Fail often and fail early with incremental development; working software is the primary measure of progress.
- Enforce Face to face communication with team members and stakeholders.
  - Delegate a customer representative to act on behalf of the opinions of stakeholders.
  - Customer opinions determine the direction of the next sprint / workflow.

- Daily standup: 15 minute session where members collectively review how they are progressing towards their goal, and note criticisms and improvements.
- Focus on quality of architecture:
  - Continuous integration.
  - Unit testing, test driven development.
  - Pair programming.
  - Enforcing design patterns onto the architecture.

1.2 *Where is ASD appropriate?*

- Complex systems with dynamic, non deterministic and non linear characteristics, since accurate estimates and stable plans are hard to get in early stages.
- In a desire to reduce the *leap of faith* that is needed before any evidence of value can be obtained, in system development.

Part II

Legal, Ethical and Professional Conduct

There are a number of *standards* which highlight a set of axioms by which software engineers should operate, to encourage ethical and legal behaviour:

2 *ACM (Association for Computing Machinery) [58,68]*

ACM are the worlds largest computing society which provide activities, guidance and principles to support professional growth.

2.1 *General Ethical Principles:*

- Contribute to society and to human well being, acknowleding that all people are stakeholders in computing.
  - Avoid harm.
  - Be honest and trustworthy.
  - Be fair and take action not to discriminate.
  - Respect the work required to produce new ideas, inventions, creative works, and computing artifacts.
  - Respect privacy.
  - Honor confidentiality.

3 *IEEE (Institute of Electrical and Electronics Engineers) [69,74]*

IEEE is the worlds largest techincal professional organisation which enforces 8 practices which software engineers should take with regards to ethics and professional practices.

3.1 *IEE Code of Ethics.*

- Upload the highest standards of integrity, responsible behaviour and ethical conduct in professional activities.
- Treat all persons fairly and with respect, to not engage in harassment or discrimination, and to avoid injuring others.
- Strive to ensure code is upheld by colleagues and co-workers.

4 *BCS [75,77]*

BCS is a Chartered Institute for IT, and is known as an independant professional body. BCS enforces four key principles:

4.1 *General Ethical Principles:*

- You make IT/ software for everyone.
  - Address wider societal issues, and upholding standards by conducting yourself professionally and fairly at all times, to make software accessible.
- Show what you know, learn what you don't.
  - Keep on learning, continuously learn and develop yourself. You must know the boundaries of what you do or don't know.
- Respect the organisation or individual you work for.
  - Work with due care and diligence while taking personal and collective responsibility for your actions; musn't work with conflicts of interest.
- Keep IT real. Keep IT professional. Pass IT on.
  - Uphold the reputation of the profession and encourage others to grow.

5 *Research Ethics [158,159]*

5.1 *EIRA [https://moodle.bath.ac.uk/pluginfile.php/1469988/mod_resource/con*

Engaging with research involves ethical, legal and professional considerations.

- The University of Bath requires the EIRA1 electronic form to be completed and submitted for review for all projects that involve human participants.

5.2 *Scientific Integrity*

- We are expected to act with **Scientific Integrity**, which enforces the following principles:
  - *Honesty*: The work you present must be clear and your intentions must be accurate. Interpretations of data should be clear, and justifications of data; acknowleding sources, and honesty in the farming of data.
  - *Rigour*: Research to be rigorous and explore and exhaust all nodes.

  - *Transparency and open communication*: highlight limitations of studies and negative results. Must be transparent in your methodologies, and share criticisms as part of the research process.
  - *Care and respect*: to all beneficiaries of the experiment, including animals, environments and cultural objects.
  - *Accountability*: Individuals are held to account when their behaviour falls short of the standards we set.

5.3 *Misconduct types:*

- *Falsification:* Manipulating research, equipments or process; changing data to fit narratives or other goal without any justification. Remember, negative and null results are valid; it is a skill to show negative results in a positive light so as to learn from them.
- *Plaigarism:* Using other peoples work and ideas without giving proper credit to the original source. Violates the rights of the authors to their intellectual output.
- *Fabrication:* Making up results and recording them as though they were real.

5.4 *Key Principles of Human Research*

5.4.1 *Ethical Principles in Human Research*

In collecting data from people, there are some basic ethical principles to consider:

- *Respect for persons:* Individuals should be treated as autonomous agents (they can make their own decisions and choices), persons with diminished responsibilities are entitled to protection.
- *Beneficence*: The obligation to do no harm, and to maximise the benefits and reduce the risk.
- *Justic*e: Do no not deny benefits or impose burdens of research to people without good reaosn.

5.4.2 *Informed consent*

The participants should choose what does and does not happen to them. There are three core elements to informed consent.

- *Information*: Participants should have access to the risks or benefits of the research; the ability to ask questions and withdraw at any time.

- *Comprehension:* Information must be provided in a coherent manner which would not curtail the participants ability to make an informed decision. It is your responsibility to ensure that they have comprehended the information.

- *Agreement to Participate:* Valid if given voluntarily and not through coercion or threat, or undue influence such as an improper reward (eg a bribe).

5.4.3 *Withdrawal*

- Participants must have the right to withdraw:
  - After having read the information sheet.
  - Part way through the study
  - After the study;

- When working with anonymised data, it may not be possible to extract an individuals data. Participants must be made aware of this, and depending on how the data is anonymised, you may want to consider a *grace period* where you can give participants to withdraw their data, after which the data will be anonymised.

5.4.4 *Data*

- Only collect data that is absolutely necessary.

- Clearly tell participants what you are recording and why. You must provide consent if you use images in publications and reports.

- Anonymise the data if possible (pixelating images).

- Secure storage of data
  - All participant data should be stored securely on the University drives, access files with files.bath.ac.uk remotely.
  - All paper data like consent forms should be stored in locked filing cabinets with clear labels.

5.4.5 *Recruitment of Participants*

- Use emails, social media, posters, word of mouth.

- You musn't coerce or pressurise people into taking part. Compensation should be to compensate their time, not entice them.

- Must provide fundamental information of the study such as:
  - Study title

  - What will be involved, and duration of study.
  - Criteria for inclusion (eg, healthy adults 18+)
  - Compensation for their time information.
  - Contact information.

5.4.6 *Participant Impairments*

You must consider the impairments of your participants of the study as the ethical process will take longer and will be more complicated. Consent could be more problematic if under 18 / those with diminished responsiblity.

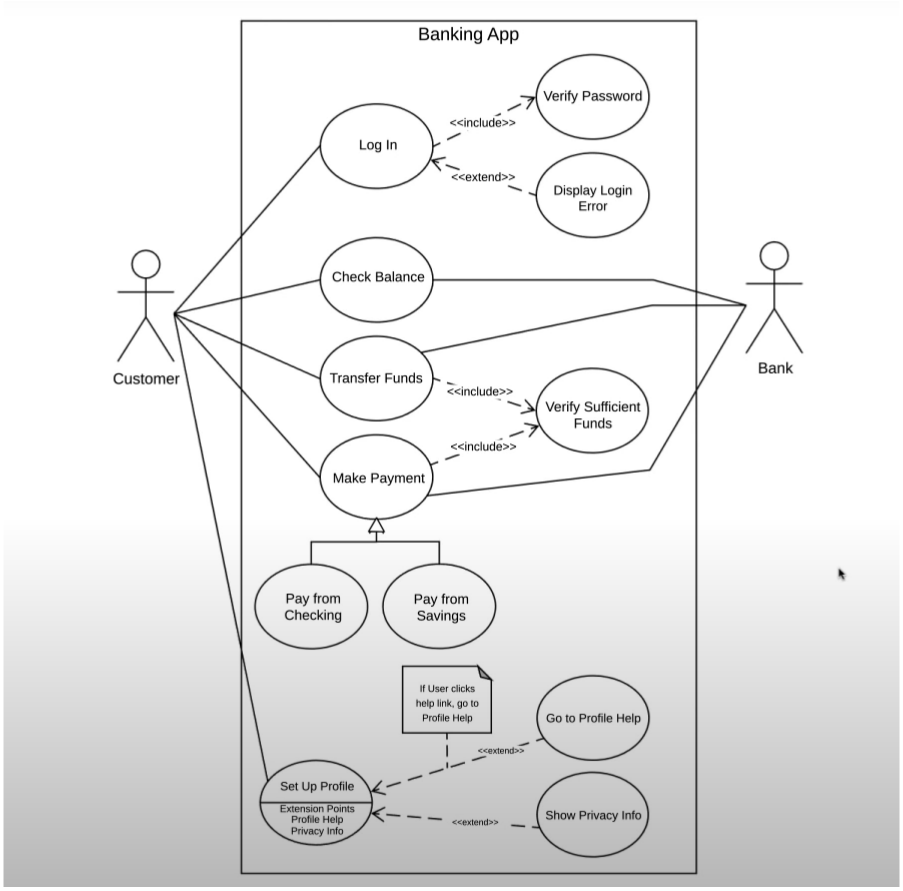
5.4.7 *Debriefing*

- Thank the participant for their time
- Provide participants with a full explanation of what was being tested.
- What were the hypotheses and what were they based upon.
- Explain how the results will be used.
- Where they can find more information.
- Reminder of their right to withdraw.

Part III
UML Models

6 *UML Case Diagrams*

- All application events are inside a box. All external interactions which produce these events are outside of the box.
- **Actors** produce events, in which case a line is drawn from the actor to the event(s).
- **Events** are drawn in circles.
- **Event chains** are defined by:
  - <<include>>, which $\Longrightarrow$ the next event.
  - <<extend>>, which could be $\Longrightarrow$ the next event.
- Events can be defined further by establishing a hierarchy which explains the proper subtypes of said event.

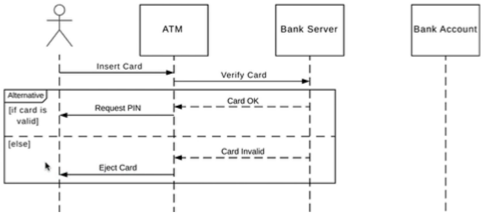
- 

7 *UML Sequence Diagrams*

UML Sequence Diagrams are a type of UQL diagram show how classes in code interact with each other, in order, namely the sequence of events. Used to document processes and understand the requirements of a new program.

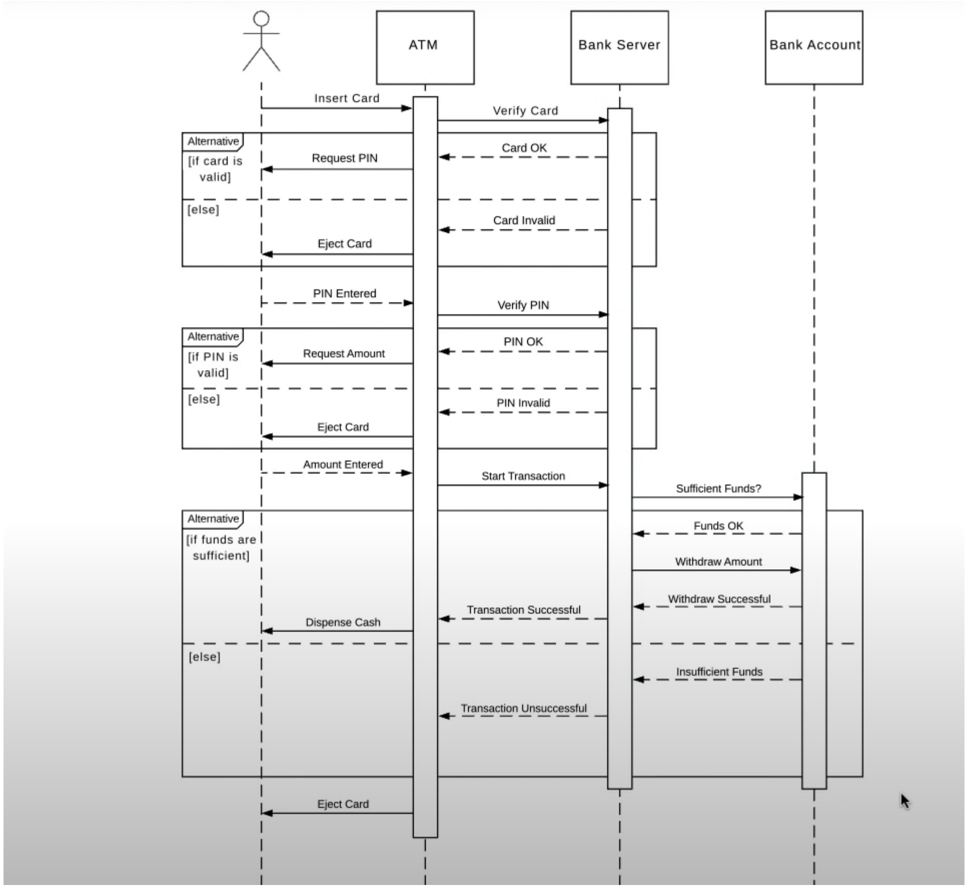
- **Actors** interact with **Objects.** Objects are placed in sequential order, and are in squares.
- **Lifelines** are vertical dashed lines which show the existence of the object over time (time axis is vertical).
- **Messages** show the messages being sent between objects. These are indicated with solid lines.
  - **Return Messages** are dashed lines and are responses to messages.

- **Alternative Frames** are basically conditional branches, which symbolises a choice between two message sequences. It is encapsulated in a box around the target message sequences.

  –

- **Activation boxes** show when and how long objects perform processes and go along the lifelines.

-

8 *Additional Resources*

- https://www.uml-diagrams.org/uml-25-diagrams.html

Part IV
Application Flow

9 *Human Centred Design*

9.1 *What is Human Centered Design?*

Human centered design is an interative design process, in which designers focus on the users and their needs in each phase of the design process. Commonly used in design, management, and engineering frameworks that develops solutions to problems by involving the human perspective in all steps of the problem solving process.

9.2 *Drawbacks of Human Centered Design*

Most people do not know what they want; as such, it would be difficult to iteratively design an application when the suggestions in question may be contradictory or incomplete.

9.3 *How can one enforce Human Centered Design? The Design Thinking Model (Five Step Process)*

Consider the ***Design Thinking*** five-stage model, proposed at the Institute of Design at Stanford.

9.3.1 ***(1)Empathise*: To empathise with another human being is to understand and share their feelings.**

We can achieve empathy through:

- ***Observing:*** view users and their behaviour in the context of their lives.
- Keep asking why when asking questions.
- Encourage stories and get worldviews.
  - Pay attention to non verbal cues such as body language and emotion.
- ***Engaging Stakeholders***:
  - ***What is a stakeholder?*** A person, group or company that is directly or indirectly involved in the project, and wo may affect or get affected by the outcome of the project.
  - ***How can we engage stakeholders?*** Interact with stakeholders through encounters. One should engage early and at every stage; this will help scope and guide as well as validate our progress towards the project goals. We can do this by defining a ***Stakeholder Engagement Plan (SEP).***

- ***Stakeholder Engagement Plan (SEP)***
    - ***Define Context***
        - Describe the problem and proposed solution and inform stakeholders of this context. Describe the system features and suggest target users and details of how and where the system will be used.
    - ***Define Engagement Goal***
        - What do you hope to gain from engaging your stakeholders?
    - ***Define your method of engagement[115,], [131,].*** These could include:
        - ***1 on 1 interviews***
        - ***Focus Groups[134,***
        - ***Surveys***
        - ***Ethnographies***
        - ***Diaries***
    - ***Define your location and profile.***
        - Where will the engagement take place?
        - Who will be involved (including team members)?
        - What will be the role of the stakeholders, and which of their characteristics make them sutiable for this engagement?
- ***Stakeholder Requirements[168,]:*** Now that we have engaged our stakeholders and gathered some data, we must now produce functional and non functional requirements for our system.
    - ***Functional Requirements:*** General system constraints, demands from the systems in terms of quality, UI, and performance. Services features and functions to provide, commonly relate to inputs and outputs. More low level and specific in terms of event chains.
    - ***Non Functional Requirements:*** General properties of the system and its behaviour, they aren't features of the system but are required characteristics or qualities (describing the system from a high level)
    - ***Requirements Engineering[178,], [183,], [187,]:*** To produce these requirements, refer to the seven step process. We can ***collaborate[187,]*** in gathering these requirements. We can organise these functional and non functional requirements using ***MoSCoW prioritisation[193,]***
    - ***Analysis of Requirements[194,]***
        - ***Explicit Analysis of Requirements:*** One could use ***Use Cases[209,]*** / UML , or Scenarios / Storyboards to analyse the requirements.

        - ***Negotation of Requirements[203,]:*** Negotiate on the requirements with the stakeholders to compromise / get a win-win result.
        - ***Validation of Requirements[204,]:*** Verify that each requirement is consistent / conforms to the system.
    - ***Immerse:*** Experience what our users experience.

9.3.2 *(2)Define*: Unpacking our findings from Empathy, into compelling needs, insights and a meaningful challenge.

We are laying concrete definitions and focusing on a problem that the user(s) may experience. Our goal is to develop a deep understanding of the users and the design sapce, and to come up with an actionable problem statement. We can analyse our Empathy findings by the following:

- ***Needs:*** What are some needs that we have discovered?
    - These needs could be physical and or emotional.
    - ***Insights:*** What are some insights that we have found interesting?
        - Something that may not conform to the norm; this could give you a leverage or a unique selling point to your application. We can further leverage these insights to tailor our applications to our users and promote equity.

9.3.3 *(3)Ideate:* Generation of Ideas.

Now that we have an actionable problem statement, as well as a rough reference on the user insights and problems, we can begin to generate ideas. The goal now, is to explore a large solution space and develop a large, diverse set of ideas. From this tree of ideas, we can begin to prune and build prototypes to test with users. The ideate process is as follows:

- ***Sketching***: Go for volume.
    - ***Pruning:*** We can narrow our selection down by:
        - Voting, where the team iteratively votes for the best ideas across an interval.
        - Restrict the ideas to four categories, and elect the appropriate idea(s) within each category:
            - The most rational choice
            - The most daring
            - The most likely to delight
            - The long shot
    - ***Feedback on pruned ideas:*** Find feedback from participants and your team-members. Criticism is essential. Then, iterate and refine your ideas based on this feedback.

9.3.4 *(4)Prototyping and different prototype styles[262,]:* Making early samples, or models of a product built to test a concept or process.

Now that we have modelled our requirements for the system, we must produce a prototype and translate these requirement models into ***design models[222,]***.You can create a digital or physical prototype of your solition, which the user or team member can engage and interact with. Here is what you should consider when commencing the translation:

- ***Software Architecture[229,] and its different styles [247,]***
- ***Design Patterns[231,]***
- ***Seperation of Concerns[232,***
- ***Modularity[233,]***
- ***Encapsulation and Information Hiding[235,]***
- ***Functional Independance [236,]***
- ***Stepwise Refinement[238,]***

To ensure that the UX is friendly and responsive, we need to consider ***Interaction Design[266,]***

9.3.5 *(5) Testing*[318,]: An iterative process in which we refine our solutions and make them better.

Now that we have developed our prototype, it is time to iteratively test and refine it.
Here are the fundamentals of testing:

- ***Validation and Verification[320,]:*** You will be iteratively performing validation and verification tests throughout the testing phase.
- ***Test Driven Development[324,]:*** This is an algorithm which involves knowing what tests before writing the code, involving:
  - ***Creating a test plan[325,]***
    - ***Creating a test strategy[337,]***
    - ***Writing Test Cases [326,]***
      - There are different ***levels of testing[338,]***, and ***different types of testing (manual vs automated testing)[345,]***
  - Developing your code.
  - Testing your code.

9.3.6 *(6 - ADDITIONAL) Evaluation of the Application*

You should consider the following points:

- ***Where to evaluate? [357,]***
  - ***Controlled Settings:*** Usability and reseach labs for usability testing and experiments.
  - ***Natural Settings:*** Online communities and products used in public places.
  - ***Any settings not directly involving users[361,]***
- ***Evaluation Methods[365,]*** describe how exactly the data from the evaluation experiment will be gathered. There are two types of evaluation methods:
  - ***Quantitative Evaluation:*** Data are numbers, good for comparing solutions or test aspects of the systems. More mathematical with measurable metrics, scales and changes.
  - ***Qualitative Evaluation:*** Data is typically text. Good to explore user needs and problems; analysis of data for themes and patterns. More psychological, involving interviews and open ended questions.
- ***Analytic Evaluation[366,]:*** Involving experts, targeting ***usability***, which is measured through mathematical / heuristic analysis. Usability is the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use.
  - Uses ***Heuristics[379,]*** to check whether the UI elements conform to usability principles.
    - These heuristics could be questions or ratings.
    - Refer to ***Nielsen's 10 UI Heuristics[379,]:*** allows one to measure usability in application design.
      - To conduct a ***Heuristic Evaluation[386,]*** there are three steps.
      - ***Briefing Session:*** experts are briefed about the goal of the evaluation, and about the typical users and their key tasks.
      - ***Evaluation Period:*** Each expert works seperately. First pass for a feel of the flow of interaction and system in general. Second pass focusing on specific features / interace elements. Experts can assign severity ratings to each feature.
      - ***Debriefing Session:*** Experts work together with the design team to discuss findings, their prioritsation and suggestions for solutions.

  - Uses ***Cognitive Walkthroughs[389,]*** to simulate users problem solving, and checking whether the user would know what to do at each step.:
    - Task based approached of inferring users problem solving / mental process when they explore an interace.
    - Involves experts and members of the design team. Focuses of ease of learning of the system.
    - There are four stages of ***conducting a cognitive walkthrough[390,]:***
      - ***Preparation:*** Identifying the goals you want to examine, the characteristics of typical users, and tasks to be completed to accomplish these goals.
      - ***Briefing:*** Designer presents the design, and expert is told about the population, task details.
      - ***Evaluation:*** Experts walk through action sequences for each task. They answer the ***four questions[391,]*** and note problems.
      - ***Debriefing and Revision:*** A summary of the results are compiled and insighed checked with users; designed is criticised and iteratively improved.
  - ***Measuring Usability[399,], [418,]:***
    - ***Analytical Methods[395,]:*** To use as metrics, such as IO tracking etc, ***Fitts Law (Predictive Modelling)[396,], System Usability Scale (SUS)[420,]***

- ***Empirical Evaluation[367,], [408,]:*** Involving users, the experiment. Based on anecdotal, experimental evidence, not mathematical. In conducting an empirical evaluation, the following steps must be conisdered:

  1. ***Choose Research Questions[410,]***
  2. Specify Methodology
     - ***Definition of tasks for the users[413,]***
     - ***Definition of the variables in the application[415,]***
     - ***Specify the hypothesis based on the changing variables (your expected outcome)[416,]***
  3. ***Conduct a Pilot Study:*** A small test study.
  4. ***Use a script to collect data[424,]***
  5. Analyse data to find answers, and discuss.

- ***Data Reliability[369,]:*** You must consider whether or not the data from the evaluation is reliable, valid, the scope of the data, and the presence of any ***Biases and Limitations[370,]***