THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

DEPARTMENT OF
COMPUTER SCIENCE
Te Tari Rorohiko

2025

Contributors

J. Turner

R. Mercado

V. Moxham-Bettridge

J. Kasmara

INTRODUCTION TO THE SOFTWARE DEVELOPMENT LIFECYCLE

In software development, projects which have a clear structure and are well managed tend to be more successful. The Software Development Life Cycle (SDLC) allows us to follow an overall structure for a project. It is based on "best practices" when creating new systems. In this session we introduce the SDLC and will use this structure for subsequent sessions in order to develop an application for commuters.

The Software Development Lifecycle

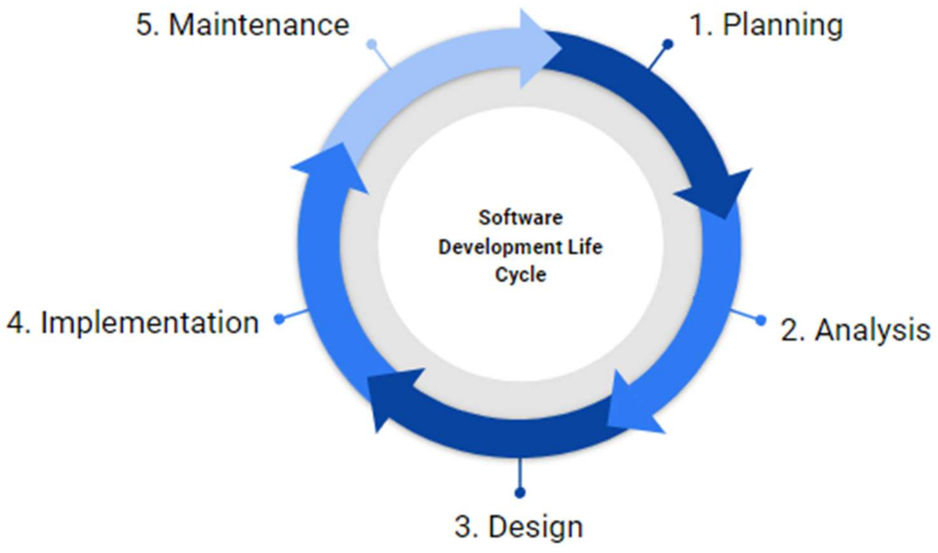

Figure 1: The 5 stages of the SDLC

The SDLC consists of 5 development phases: planning, analysis, design, implementation and maintenance.

Planning

Planning involves identifying the need for a new or enhanced system. We wish to determine what problem a client is facing and how we can address that problem. We do not want to build software for the sake of building software, therefore we will need to determine the benefits of the new system for the client so they can determine if the investment is worthwhile. We also need to specify how long it will take to get a first implementation.

Problems should be detailed and clearly linked to the client's needs, for example:

Because the client organisation is maintaining billing information for more than 100 of its clients, it is:

(a) Spending an exorbitant amount of its staff time on creating and sending out invoices (currently 50 hours per month), and
(b) Staff members are misplacing and not sending out paper invoices (and so estimate losing $400 per month in income)

Here we identify the client's specific problem and detail the issues that need to be addressed by the software we will develop. Specifically, the software must reduce the number of hours required to create and send invoices, in addition to ensuring that invoices are sent out correctly to avoid a loss of income.

Note that problems are NOT "*Client has a paper-based system and needs a database"* which is an oversimplification of the problem. It is important when developing software that we identify the client's needs and then write software to meet those needs.

With a clearly defined problem we can define the benefits of the software to be developed. There are two types of benefits that we consider: tangible and intangible. A tangible benefit can be measured, for example:

This system will reduce the receptionist's workload by two hours a week. As the receptionist works 26 weeks at $17.70 an hour, this saves $920.40 per year.

An intangible benefit cannot be measured, for example:

Employee satisfaction will be increased.

More professional appearance to customers.

While they cannot be directly measured, intangible benefits can be important, but they are usually not important enough to justify the creation of a new system.

Analysis

The analysis phase consists of two sub-processes:

- What are the requirements of the system?
- Out of those requirements, which are important, redundant, etc.?

This is NOT a detailed design of the system, simply a stage which allows you to figure out "what" the system should do for the client. We consider two types of requirements, functional and non-functional requirements. There are several ways requirements can be handled, here we present one way as an example.

Functional requirements describe the functionality that is required of a software system. Examples include: calculations, technical details, data manipulation and processing, any specific functionality for end users, and business processes/tasks that the software must support. For example:

- The system will allow a user to create an invoice.
- The system will allow a user to add client information to a database.
- The system will allow the user to produce a report.

Non-functional requirements are criteria the system and user interface must satisfy to be useful/successful/acceptable. They are based on the "ilities" of software development (e.g. accessibility, affordability, customizability, usability, flexibility and so on). Non-functional requirements are usually constraints, for instance how fast, how safely, how securely, how efficiently etc. must a particular task be carried out by the system? For example:

- Production of the sales report must take less than 2 minutes.
- Invoices should be sent within 12 hours of creation.
- The pump must not dispense a deadly amount of medication.

Design

This is the part of software development where requirements are converted to system specifications. All aspects of the system are designed such as the user interface, program flow, database design and so on. Models of the system may be created to help define how the system should work. We will look at the types of models that can be used in a later session.

Implementation

Using the specification we can determine how the system should work and create an implementation which satisfies that specification i.e. we create the designed system. The specifications are turned into a working prototype that can be used and tested. This phase includes coding, testing and installation as well as user support, such as documentation on how to use the new system.

Maintenance

Once a system has been created it requires ongoing maintenance as users of the new system will often find issues and bugs. This is not necessarily a "separate" phase but encompasses the ongoing iteration of the 4 previous phases. This allows systems to continue to evolve and grow with the client or business, keeping the software relevant to current practices.

Understanding the customer

Watch the following comedic take on understanding the customer:
https://www.youtube.com/watch?v=BKorP55Aqvg

It is important when designing software that we understand what the client wants and that we communicate clearly whether or not this is possible. While the SDLC gives us an idealistic view of what software development should look like, in reality this is not the case. For example, consider the following image:

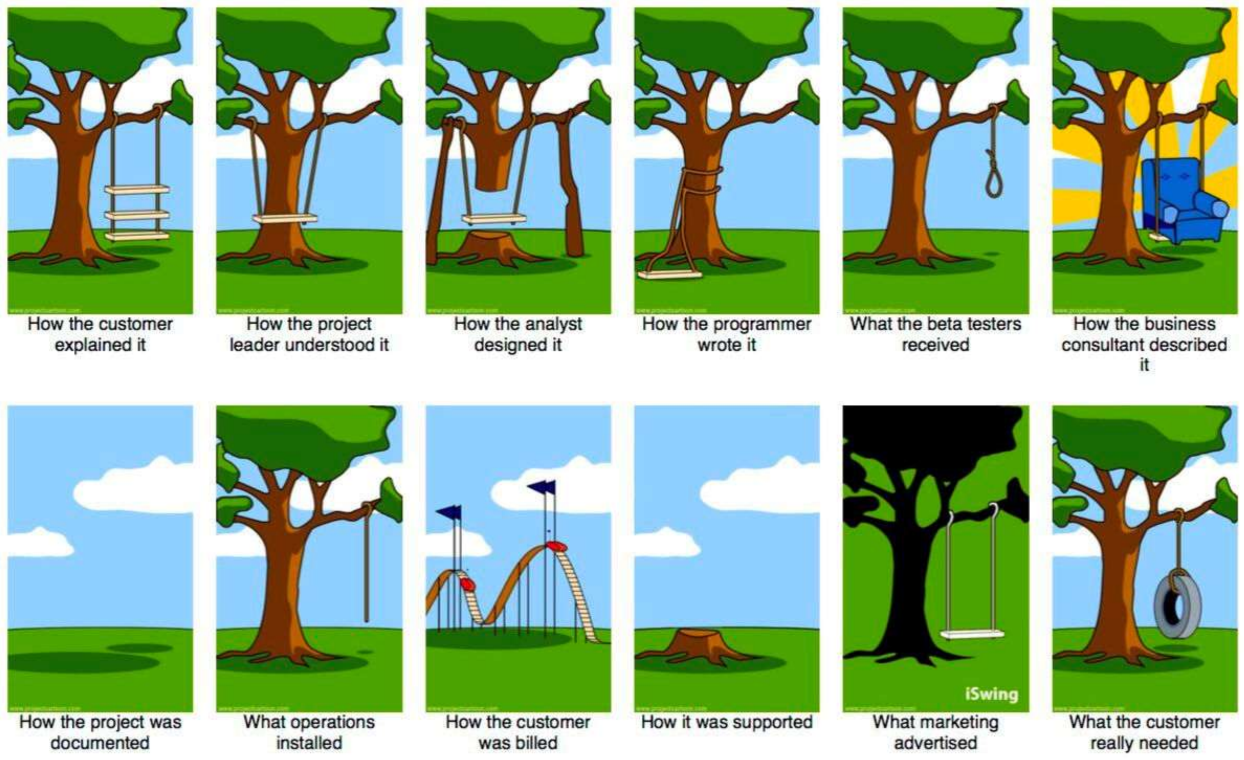
How the customer explained it
How the project leader understood it
How the analyst designed it
How the programmer wrote it
What the beta testers received
How the business consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
iSwing
What marketing advertised
What the customer really needed

Figure 2: The SDLC from different perspectives

This illustrates in a simple way how difficult it is to create software. There are different groups of stakeholders each with different perspectives, and within those stakeholders are technical and non-technical people. That is, there are those who are domain experts, such as: software developers, engineers, testers; and those who are not, such as: clients, UX designers, marketing professionals and so on.

Sometimes the customer may describe what they want and it may not be what they really needed, when this is then churned through each of the different perspectives, we often end up with something that isn't what we expected at all. This is why it is important to have clear repeatable processes for creating software.

L2 Activities

Over the course of the term you will be working with the SDLC. While you won't have a "client" necessarily, you will have the information you need to design a software system. For the first half of the program you will work with toy examples and practice some of the techniques required for software development. In the second half you will build the software using Visual Studio and the C# programming language.

Exercises: Planning

You have received the following email about a new software development project for a company called *Parkway Commute*:

11/04/22

Kia ora,

Following on from our conversation the other day, we have a large data source from surveys of our employees describing their commuting habits to and from work. We want to use this data to help us with carbon reduction planning. We need a dashboard that will allow us to explore the data that we have collected and enable us to make planning decisions. We want to encourage our employees to reduce their carbon footprint by using single-car use alternatives such as cycling, walking, bus and so on.

We look forward to hearing your proposal.

Ngā Mihi,
Joe Bloggs
Parkway Commute

Recall, that as part of the planning phase we need to answer the following questions:

1. Why should the proposed system be built?

2. What will the proposed system do?
3. Who will be involved in using the system?
4. Where will it fit into the existing system(s)?
5. When will all/part be ready for use?
6. How will it function, are there any constraints?

In a real world setting the best way to answer these questions would be to discuss the requirements with the client. You might use interviews, questionnaires, or observation to gain an understanding of current practices. However, given this is a fictional setting, assume the following back and forth has occurred:

12/04/22

Kia Ora Joe,

Thanks for reaching out. I have a few questions based on what you've said, these are as follows:

- *Could you send me a copy of some sample data?*
- *What information do you expect the dashboard to show?*
- *What types of planning decisions would you base on this?*

Look forward to hearing from you.

Ngā Mihi,
Carlene Webb
Software Solutions LTD

13/04/22

Kia Ora Carlene,

Thanks, see my answers below.

Kia Ora Joe,

Thanks for reaching out. I have a few questions based on what you've said, these are as follows:

- *Could you send me a copy of some sample data?*

Yes, find sample data attached.

- *What information do you expect the dashboard to show?*

It would be good to see each individual's carbon footprint, what area they are travelling from, as well as an overview of the company as a whole.

- *What types of planning decisions would you base on this?*

We plan to offer spot prizes etc. for those that reduce their carbon footprint, so we will continue our surveys over time. We also may offer minibus or van services from popular suburbs.

Look forward to hearing from you.

Ngā Mihi,
Carlene Webb
Software Solutions LTD

Excerpt from attached file *employees.csv* file:

ID	First Name	Last Name	Role	Home	Work
233	Mary	Bennett	Secretary	23 Cameron Road, Tauranga	148 Durham Street, Tauranga
453	Jim	Carter	Engineer	4 Dee Street, Mount Maunganui	148 Durham Street, Tauranga
675	Richard	Walker	IT	26 Tweed Street Mount Maunganui	52 Miro Street, Mount Maunganui
…					

Excerpt from attached file *commutes.csv* file:

EmployeeID	Arrive By	Trip Time	Depart By	Trip Time
233	9:00am	10 minutes	5:30pm	20 minutes
453	8:00am	45 minutes	4:00pm	50 minutes
675	8:30am	50 minutes	5:00pm	60 minutes
…				

Excerpt from attached file *travelModes.csv* file:

EmployeeID	**TravelMode**	**EmissionsType**
233	Car	Petrol
453	Car	Hybrid
675	Bus	Don't know
…		

14/04/22

Kia ora Joe,

Thanks for the information. In terms of Travel Modes and Emissions Types, what is the exhaustive list of options?

Ngā Mihi,
Carlene Webb
Software Solutions LTD

14/04/22

Thanks for your email. We don't have any predefined categories but here's what was included in the spreadsheet:

- *Travel modes: car, bus, walk, bike, car pool*
- *Emissions types: Petrol, Diesel, Hybrid, Electric, None, Don't know*

Ngā Mihi,
Joe

Using the above information, prepare the following proposal:

- What is the problem?
- Who is the company and what do they do?
- What is your proposed solution?
- What are the tangible and intangible benefits?

ANSWERS:

What is the problem?
Parkway commute wishes to understand the carbon footprint of its employees commutes so that it can make planning decisions to reduce carbon emissions.

Who is the company and what do they do?
Joe has not provided enough information, we would need to do further research.

What is your proposed solution?
To create a piece of software that will allow Parkway Commute to explore their survey data by company and by individual. The software will compute carbon emissions for each employee. In addition, it will display trips by suburbs/location to determine high density commutes to enable decision making around Parkway Commute supplied carpooling.

What are the tangible and intangible benefits?

Tangible:
Again, we need further information from Joe. But you could imagine:

- Parkway commute's current carbon footprint of XtCO_2 will be reduced by X amount.

Intangible:

- Improved commuting experience for employees will make them more productive and more keen to come into work.

Summary

In this session we introduced the SDLC and began the planning process for the software that you will create this term. Next week we will move onto the analysis phase and determine the software requirements for the project.

Useful Resources

- SDLC Overview: https://www.tutorialspoint.com/sdlc/sdlc_overview.htm
- SDLC in 9 Minutes: https://www.youtube.com/watch?v=i-QyW8D3ei0
- Lane, D. (2011). A Beginner's Guide to the Software Development Life Cycle. In The Chief Information Officer's Body of Knowledge (pp. 129-136). Hoboken, NJ, USA: John Wiley & Sons. https://www.researchgate.net/publication/316422814_A_Beginner's_Guide_to_the_Software_Development_Life_Cycle