**For**

**Teaching**

**Projects**

Table of Contents

[2. BACKGROUND 3](#_Toc24208660)

[Learning Theories 3](#_Toc24208661)

[Learning Outcomes 4](#_Toc24208662)

[Instructional Design Models 5](#_Toc24208663)

[Reviews of Existing Software Systems 7](#_Toc24208664)

[3. DESIGN 8](#_Toc24208665)

[Methodology 8](#_Toc24208666)

[System Architecture 9](#_Toc24208667)

[System Overview 10](#_Toc24208668)

[Front-Tier 10](#_Toc24208669)

[Middle-Tier 10](#_Toc24208670)

[Back-Tier 10](#_Toc24208671)

[5. EVALUATION 11](#_Toc24208672)

[Nielsen’s Heuristics 11](#_Toc24208673)

[eLera’s LORI 12](#_Toc24208674)

[Evaluation of Software 13](#_Toc24208675)

# 2. BACKGROUND

## Learning Theories

How do people learn? We don’t really know, but we have a few theories, and here are the three most popular theories. So it’s not that you have to choose any one of these; they are all right, so a mixture of these theories of learning is what we should assume.

|  |
| --- |
| ***Behaviourism****:* This theory says you shouldn’t worry about how people think, the way they learn is through behaviour, so if you want to learn something, repeat it over and over again until you have memorised it, and that’s the start of learning. As well as repetition, another way to learn is reinforcement, so desired behaviour is rewarded, while the undesired behaviour is not rewarded.   * For Your Project   + Multi-choice questions that the learners can take and retake   + Videos that they can replay.   + Repeat the same content, or at least link back to things they have seen before |

|  |
| --- |
| ***Cognitivism:*** This theory says that the study of the brain is the most important way to understand how people learn. Understanding cognition, attention, short-term and long-term memory helps you understand learning. Memory is structured into schemas, so integrating new learning into your memory requires that you restructure existing schema to incorporate that information.   * For Your Project   + Use analogies and metaphors where possible   + Provide hyperlinks to useful content   + Use mnemonics, concept maps, and advanced organizers |

|  |
| --- |
| ***Constructivism:*** This theory says knowledge and learning are constructed, individually or as a group, student create meaning on the basis of past experience, and their reflection on those past experiences. Therefore different people have different versions of the truth, and they must be actively involved in their learning, and must interact with other to reach an agreed truth.   * For Your Project   + Provide a discussion board or Wiki for students to share their views and research   + Create WebQuests for the students   + Create activities that encourage reflection |

## Learning Outcomes

Learning outcomes are created by the educator, and they are short, simple sentences that outline what a learner is expected to know after a particular learning experience. So if the content being taught is related to education, a statement of learning outcomes might be:

|  |
| --- |
| On Completion of this module, the learner will be able to:   * **Identify** and **review** a wide variety of learning and teaching methods * **Discuss** the theories of learning that underpin their teaching approach * **Develop** effective and efficient self-directed study skills |

So the important thing to note here is that we use action verbs to describe what we expect the learners to be able to do. When possible we try to avoid using the words ‘know’, ‘understand’ or ‘appreciate’ as these tend to be vague and hard to quantify. Instead we try to think of what the learners should be able to do in order to demonstrate they have gained the required knowledge. These learning outcomes will give you clues as to how to teach the content, as well as what type of blended activities might be suitable, and what form of assessment you might use.

In 1956 Benjamin Bloom led a group of educators to develop a classification of levels of learning, moving from non-critical learning (surface learning) to critical and reflective learning (deep learning). The lowest level of learning, when the learner is taking in the information, and trying to remember as much of it as possible without any critical reflection is Level 1 (called the “Knowledge” level), the highest level is Level 6 where the learners have deeply engaged with the material is Level 6 (called the “Evaluation” level). Below is a description of each of the levels, and some verb examples that can be used for writing learning outcomes.

|  |  |  |
| --- | --- | --- |
| **Level** | **Description** | **Action Verbs** |
| **1. Knowledge** | Remember previously learned information. | arrange, define, duplicate, label, list, memorize, name, order, recognize, relate, recall, repeat, reproduce, state |
| **2. Comprehension** | Demonstrate an understanding of the facts. | classify, describe, discuss, explain, express, identify, indicate, locate, recognize, report, restate, review |
| **3. Application** | Apply knowledge to actual situations. | apply, choose, demonstrate, dramatize, employ, illustrate, interpret, operate, practice, schedule, sketch, solve, use |
| **4. Analysis** | Break down ideas into simpler parts and look for generalizations. | analyse, appraise, calculate, categorize, compare, contrast, criticize, differentiate, discriminate, distinguish, examine |
| **5. Synthesis** | Compile component ideas into a new whole solution. | arrange, assemble, collect, compose, construct, create, design, develop, formulate, manage, organize, plan, prepare |
| **6. Evaluation** | Make and defend judgments based on internal evidence or external criteria. | appraise, argue, assess, attach, choose, compare, defend estimate, judge, predict, rate, score, select, support, value |

## Instructional Design Models

Following on from the Learning Theories, are there any specific guidelines, or sequences of steps we should follow when designing learning content (models of Instructional Design).

|  |
| --- |
| ***ADDIE:*** This model says when developing teaching materials, do the following:  *ANALYSIS*: Figure out what the students need to know  *DESIGN*: Design the content based on the needs, use storyboards and personas  *DEVELOPMENT*: Develop the content  *IMPLEMENTATION*: Teaching some students with the content  *EVALUATION*: Check how well the teaching went (EVALUATION).  This model is a lot like the Software Waterfall Lifecycle.   * Molenda, M. (2003). In search of the elusive ADDIE model. Performance improvement, 42(5), 34-36. |

|  |
| --- |
| ***Gange’s Nine Events of Instruction:*** This model says consider nine things when writing a lesson:  (1) *Gain attention of students* – Maybe create a splash page  (2) *Inform learners of objectives* – Tell the students what you are going to teach  (3) *Stimulate recall of prior learning* – Try to link the content to something they already know  (4) *Present the content* – Teach them the content  (5) *Provide “learning guidance”* - use of examples, non-examples, case studies, graphical representations, mnemonics, and analogies  (6) *Elicit performance (practice)* – Get them to do a quiz  (7) *Provide feedback* – give formative feedback if possible  (8) *Assess performance* - Do a final assessment  (9) *Enhance retention and transfer to the job* – Give them a PDF that summarizes the entire content of the course.   * Gagne, R. M., & Briggs, L. J. (1974). Principles of instructional design. Holt, Rinehart & Winston. |

|  |
| --- |
| ***Elaboration Theory:*** This model says there are eight things to consider when designing a lesson: (1) *Organizing Course Structure* - Single organisation for complete course  (2) *Simple to complex* - Start with simplest ideas, in the first lesson, and then add elaborations in subsequent lessons  (3) *Within-lesson sequence* - general to detailed, simple to complex, abstract to concrete  (4) *Summarizers* - content reviews presented in rule-example-practice format  (5) *Synthesizers* - Presentation devices that help the learner integrate content elements into a meaningful whole and assimilate them into prior knowledge, e.g. a concept hierarchy, a procedural flowchart or decision table, or a cause-effect model.  (6) *Analogies* - relate the content to learners' prior knowledge, use multiple analogies, especially with a highly divergent group of learners.  (7) *Cognitive strategies*- variety of cues - pictures, diagrams, mnemonics, etc. - can trigger cognitive strategies needed for processing of material.  (8) *Learner control* - Learners are encouraged to exercise control over both content and instructional strategy. Clear labelling and separation of strategy components facilitates effective learner control of those components.   * Reigeluth, C. M. (Ed.). (2013). Instructional design theories and models: An overview of their current status. Routledge. |

|  |
| --- |
| ***Learning Styles:*** This model says that different people learn differently, so you need to have the same content in a range of formats to address different learner needs. A simple model is VARK:  *VISUAL*: Some students like Diagrams  *AUDITORY*: Some like to listen to lectures and Podcasts  *READ&WRITE*: Some like read and write text  *KINESTHETIC*: Some like to do practical stuff |

|  |
| --- |
| **Key References** |
| * Alessi, S. M., & Trollip, S. R. (2000). Multimedia for learning: Methods and development. Allyn & Bacon, Inc. * Clark, R. C., & Mayer, R. E. (2016). E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning. John Wiley & Sons. * Hattie, J. (2003). Teachers Make a Difference, What is the research evidence? Australian Council for Educational Research (ACER) Annual Conference on: Building Teacher Quality. |

## Reviews of Existing Software Systems

* Find 3-6 software systems or apps that do something similar
* Include a picture of the system
* Include a description of the key features
* Include 2-3 sentences on how this might inspire your system

|  |  |
| --- | --- |
| **Programming Is Easy (PIE) System** | |
|  | **Description**  This is the *Programming Is Easy* (PIE) system to help students learn how to program by presenting the students with a range of problems and providing a flowchart for them to insert different instructions into each of the steps.  Features include:   * Interactive steps, with drag-and-drop * A range of problems * Automated marking |
| **In This Project**  Include 2-3 sentences on how this might inspire your system | |

# 3. DESIGN

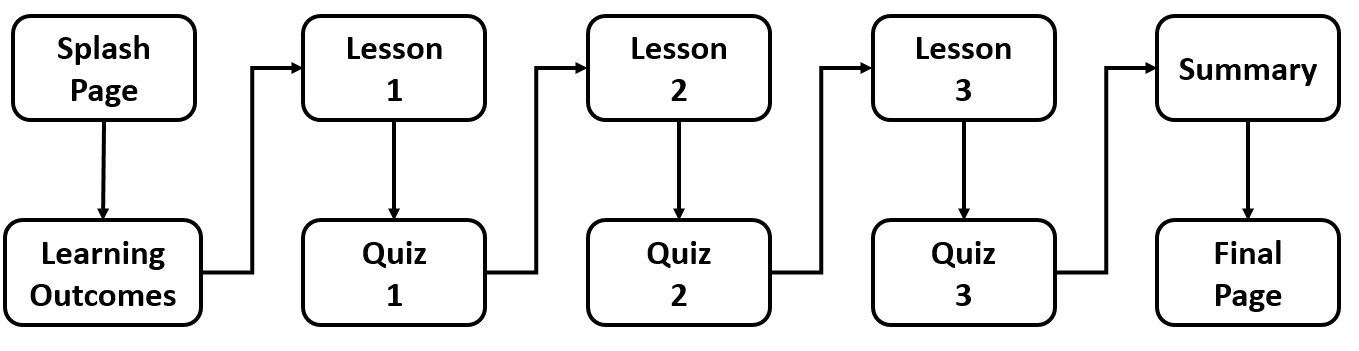
## Methodology

|  |  |
| --- | --- |
| **Waterfall Model** | [Image] |
| **Spiral Model** | [Image] |
| **V Model** | [Image] |
| **FDD** | [Image] |
| **TDD** | [Image] |
| **Scrum** | [Image] |

## System Architecture

|  |  |
| --- | --- |
| **Stand-Alone**  **(1-Tier)** |  |
| **Client-Server**  **(2-Tier)** |  |
| **3-Tier** |  |
| **N-Tier** |  |

## System Overview

****

## Front-Tier

* Paper Prototypes

Wireframes

* Use Cases - don’t just create one diagram, show an evolution of diagrams, from:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Initial Design** | **Intermediate Design** | **Final Design** |

## Middle-Tier

* JDBC
* ODBC

## Back-Tier

* ERD
* Class Diagram - don’t just create one diagram, show an evolution of diagrams, from:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Initial Design** | **Intermediate Design** | **Final Design** |

# 5. EVALUATION

## Nielsen’s Heuristics

To help evaluate the usability of the system, Jakob Nielsen create a series of heuristics (rules-of-thumb) that are probably the most-used usability heuristics for user interface design. Nielsen developed the heuristics based on work together with Rolf Molich in 1990.

|  |  |
| --- | --- |
| Title | Description |
| **1. Visibility of system status** | The system should always keep users informed about what is going on, through appropriate feedback within reasonable time. |
| **2. Match between system and the real world** | The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order. |
| 3. User control and freedom | Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo. |
| **4. Consistency and standards** | Users should not have to wonder whether different words, situations, or actions mean the same thing. |
| **5. Error prevention** | Very good error messages shows a careful design which prevents a problem from occurring in the first place. Either eliminate error-prone conditions or check for them and present users with a confirmation option before they commit to the action. |
| 6. Recognition rather than recall | Minimize the user's memory load by making objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate. |
| 7. Flexibility and efficiency of use | Accelerators -- unseen by the novice user -- may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions. |
| 8. Aesthetic and minimalist design | Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility. |
| 9. Help users recognize, diagnose, and recover from errors | Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution. |
| 10. Help and documentation | Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large. |

* Nielsen, J., and Molich, R. (1990) “Heuristic Evaluation of User Interfaces”, *Proc. ACM CHI'90 Conf.* (Seattle, WA, 1–5 April), pp. 249–256.

## eLera’s LORI

eLera is the *eLearning Research and Assessment Network* which allows educators to submit teaching content to their site, they review it using LORI (Learning Object Review Instrument) a checksheet used to evaluate the quality of e-learning resources.

|  |  |
| --- | --- |
| Title | Description |
| **1. Content Quality** | Veracity, accuracy, balanced presentation of ideas, and appropriate level of detail. |
| **2. Learning Goal Alignment** | Alignment among learning goals, activities, assessments, and learner characteristics. |
| 3. Feedback and Adaptation | Adaptive content or feedback driven by differential learner input or learner modelling. |
| **4. Motivation** | Ability to motivate, and stimulate the interest or curiosity of, an identified population of learners. |
| **5. Presentation Design** | Design of visual and auditory information for enhanced learning and efficient mental processing. |
| 6. Interaction Usability | Ease of navigation, predictability of the user interface, and the quality of UI help features. |
| 7. Accessibility | Support for learners with disabilities. |
| 8. Reusability | Ability to port between different courses or learning contexts without modification. |
| 9. Standards Compliance | Adherence to international standards and specification. |

## Evaluation of Software

One of the most useful ways of evaluating the quality of software is to ask the users. It is often best to initially get informal feedback, by asking the users either as a group or individually for feedback on the software. In addition to informal approaches, there are a number of structured techniques that can be used to elicit feedback, including the following:

|  |  |
| --- | --- |
| **Interviews** | Interviews generally start will a little small talk, followed by a brief outline of the aims of the interview. There are generally three forms of interviews:   * *Unstructured Interviews*: The interviewer has a few themes they wish to explore, but there are no fixed questions before the interviews, and the response to one question can lead to a number of follow-ups. * *Semi-Structured Interviews*: A significant number of the questions are fixed before the interview, and the interviewer has leeway in asking follow-up questions when interesting responses are given. * *Structured Interviews*: All of the questions are predetermined before the interview, and often given to the interviewee to prepare answers. No new questions can be added during the interview.   In the context of software, a semi-structured approach can yield interesting outcomes. A careful reading and analysis of the answers of several interviews can identify areas of improvement. Some researchers suggest that it is necessary to do at least five interviews on the same topic to yield a sufficiently wide range of perspectives on the topic under scrutiny. |
| **Surveys** | Surveys start with a title, and have a brief statement of its aims. Surveys are usually composed of two types of questions:   * *Closed-Ended questions*: There are a limited number of predefined answers, and the learner ticks one of the answers. This type of question is easy to administer and easy to analyse responses. They do prevent the learner from raising new issues. * *Open-Ended questions*: The questions are asked and the learner can answer however they wish. This type of question is slower to administer and slower to analyse responses. They do allow the learner to raise new issues.   Surveys can be given to users and asked to be done in their own time (this is called “self-administrated”), or the evaluator can be present and can instruct the users how to fill out the questions and answer any queries (this is called “interviewer-administrated”). |
| **Focus Groups** | Focus Groups involve taking a group of users and asking them as a group a number of questions about the software. Start by thanking everyone for coming, review the goals of the group, explain how everyone will contribute, and set the tone. Repeat what you think the group are saying in different words, and respect everyone’s opinion. Sometimes users will answer questions more honestly in a group, buoyed by other comments by other users, than if they were doing one-to-one interviews. The major concern with using focus groups is that they can be difficult to manage and sometimes the group can end up discussing irrelevant topics. Focus groups can also be done online which allows for greater participation and anonymity. |

|  |  |
| --- | --- |
| **Shadowing** | Shadowing works by simply observing the users use the technology, without asking any questions and without offering any assistance. During the observation phase the users will note any questions they have, and note any issues they encountered, and after the process is over the users can ask the evaluator about all of the questions they need answering. At the same time, during the observation phase the evaluator notes down any issues they observe with the users interaction also, and those issues are discussed after the phase is over. Ideally the observations should not last longer than 30 minutes, otherwise they will be of less value and less authenticity. Analysis of the outcomes of a shadowing activity needs to be undertaken carefully, it is a mistake to attempt to generalise the outcomes of a single shadowing activity to a whole learner group. |
| **GOMS** | GOMS stands for “Goals, Operators, Methods, and Selection rules” and is an approach for describing a user’s interaction with technology:   * *Goals*: When the user wants to do * *Operators*: Activities the user must undertake to achieve the *Goal* * *Methods*: The series of *Operators* undertaken to achieve the *Goal* * *Selection rules*: If there are a number of possible *Methods* that are possible to use to achieve a *Goal*, this tells us how one *Method* is selected over others.   Modelling technology tasks with GOMS is both time-effective and cost-effective in many practical situations. |
| **Think-aloud protocol** | The think-aloud protocol is used to understand the challenges users have when interacting with a particular technology. The process is straight-forward, the evaluator decides on a number of specific tasks (usually between 5 and 9) that a learner must do when using the specific software. The evaluator gives the set of tasks to a learner and asks them to performing the tasks while speaking out loud everything they are thinking during the tasks. Normally the evaluator records this process and does analysis on the recording after the tasks are completed. This can be a very effective way of detecting where there is potentially confusing content. If the evaluator has to give prompts to the learner during the process to help them discuss their actions in more detail, make the prompts generic, e.g. “*What are you thinking now?*” and “*Why did you do that?*” |
| **Personas** | Personas are fictional characters that can be used to represent a collection of some of the kinds of people who could potentially be using a particular software. In our case we can use a single persona to model a wide range of users. The persona is a detailed description of a specific individual that allows an evaluator to look at their software from these users’ perspectives. The details can include their name, their age, their gender, their skills, their interests, and their obstacles. Benefits that have been suggested for personas include the fact that it allows the developers to focus on the users and encourages the developer to learn more about their users and understand how they are using the software. |