Computational thinking for digital technologies: Snapshot 9

PROGRESS OUTCOME 8

An Agile approach to game design

Context

Lily's class has been working on a collaborative project that involves using a software engineering approach to design a game in Unity® software. She was learning a lot from developing the game and wanted to learn more about the ways to make good software.

Insight 1: Key aspects in developing software

I did some research into software engineering to understand why it is so important. I learnt that for a software project to be successful, it must be delivered on time, within budget and with few or no errors. Failure in any one of these three requirements can cause failures in the others and can compound into the overall failure of the project. Meeting these three requirements is often difficult in reality, due to the complexity and sheer size of many projects. When a project fails, it can be very costly, and not just in monetary terms.

For example, NASA's Mars Climate Orbiter was sent into space in 1998 to orbit Mars and observe its climate. Before the satellite managed to achieve a stable orbit, it burnt up in the Martian atmosphere. On review, it was discovered that within the software controlling the satellite's thrusters, the force required to be exerted had been calculated using English pounds rather than the industry-standard newtons.

Insight 2: Wider implications of the NASA debacle

The cost of the Climate Orbiter's software failure was far greater than the $125 million dollars it cost to build. Although NASA had a working blueprint for the satellite, producing the satellite a second time with the error corrected would be pointless, as technology had developed considerably since the initial development.

Therefore, development of a new satellite would have had to begin from scratch, which would take many years and a lot more money. Also, a costly failure such as the Mars Climate Orbiter can cause a government to lose confidence, cut funding and reduce support for similar projects in the future.

Insight 3: Comparing different project methodologies

I wanted my game development to be successful, so I decided to take on board what I had learnt from the software disasters I had researched and find a good strategy for my game design project.

Projects are usually described as a series of stages, such as analysis, design, and implementation. I was currently using a basic Waterfall process to develop outcomes – this is a linear, sequential method of working through a project with five stages: analysis, design, implementation, testing and release. Each stage is visited once and completed before moving on to the next stage.

I looked at the Agile methodology and compared its more flexible, iterative approach with the Waterfall process.

Insight 4: Using the Agile approach

I decided to use an iterative Agile approach with my team to deliver the game. We wanted to make sure we delivered the project on time and within cost (we agreed that the 'cost' would be the impact of the project on our other subjects). It was also important that the game would be playable, that it would work as intended without any major errors.

We decided to focus on short iterations – rather than conducting the whole project from start to finish, we would make small features in short periods of time. In this way, the results of each iteration could be used to adjust the overall project plan. Each iteration was like a short project in itself, which was awesome, as it meant we could work together but be assessed separately. It also allowed us to adjust goals and measure progress regularly.

Insight 5: The Scrum process

We had a large team on this project – nearly half the class! We decided to use a Scrum process and set up team responsibilities accordingly. I was the Product Owner – my role was to prioritise and define the features of the project. We also had a Scrum Master, who had to ensure that the team was fully functional and productive and followed our agreed process. We had a large Development Team, which selected the sprint backlog, organised themselves and demonstrated work results to me as the product owner.

As we worked through the process, we found out which members of the team could handle particular jobs and so we were able to better allocate tasks. Also we made sure we were always working towards a realistic goal.

Downloaded from http://technology.tki.org.nz or http://seniorsecondary.tki.org.nz/Technology/Digital-technologies
Unity® is a registered trademark of Unity Technologies.
Scrum was developed by Ken Schwaber and Jeff Sutherland.

The program and processes used in this snapshot are not officially endorsed by the Ministry of Education.
ISBN: 978-1-77669-244-6

MINISTRY OF EDUCATION
TE TĀHUHU O TE MĀTAURANGA