# Introduction

## Purpose

Independent developers Saturn Games have displayed a viable game development concept as part of an international contest. As part of a government initiative to support growth of the game industry in Canterbury an award of local and national government funding has been granted to Saturn Games as a startup fund. Prompted by the need to develop a quality product under a limited budget and with the New Zealand Public as significant stakeholders; this study was commissioned to recommend the best software development approach. The report covers an explanation of several suitable approaches, their advantages, disadvantages, risks and benefits; and recommends a particular approach as the most suitable for this product.

The report as-delivered also includes a Practice Guide, as well as Tool Guides and a Learning Package. These additional artifacts are provided to enable the Saturn Games to put the recommended approach into practice.

## Product Context

### Scope

The game itself, *Gibbous Earth*, is projected as primarily a PC (Windows) game, although other platforms such as linux, iOS and Android are also considered. The project lifecycle has been initially scoped for two years to market release and the workforce is anticipated to grow from a current team of four into a team of 45 - 50 people within the first three to four months. Approximately 40% of the workforce will be dedicated to software development/quality assurance, indicating a development team of 20-25 people. This growth indicates a highly dynamic and fluid environment, requiring robust change management and communication.

The game concept, though juvenile, is already formed and has been accepted as the premise for government funding. Development is expected to honour the initial concept, however; there is scope for the product to evolve as team members provide their experience and ideas.

### Stakeholders

Saturn Games shareholders consist of the four founding staff members, and an additional three active investors who were also involved in the funding application. Saturn Games will be accountable to the New Zealand public and must be able to show positive progress throughout development. It is expected that proof of progress will be provided at approximately six monthly intervals. As of the writing of this report it is assumed, through lack of evidence to the contrary, that New Zealand Government has not issued any regulations for game development, other than publication classification requirements (New Zealand Legislation, n.d.).

### Project Criticality

The main risk of the project is loss of investment, and there are no life-critical or financial security interests. There is no risk of loss-of-contract if project milestones are not reached. The project criticality may therefore be considered to be low.

## Agile vs Plan-driven Development

There are considered to be two main philosophies which drive software development - ‘traditional,’ plan-based development and agile development (Boehm & Turner, 2009). Both ideologies have their merits and drawbacks, described by Boehm and Turner as the difference between discipline and rigour of process and project adaptability respectively. Any software development project must assess the suitability for either of these main concepts and determine if one, the other or a mixture of both are required.

Plan-driven development evolved out of environments such as aerospace and commercial industries where a focus on documentation and processes and a change-averse environment are valued (Boehm & Turner, 2009). The advantage of this philosophy is an holistic plan for and view of a complete project and its expected outcomes - plan-driven development is predictable and can be bound by regulations and industry standards.

Comparatively, the agile philosophy eschews an upfront plan and makes smaller, feature-based, working modules of a product which is ultimately developed over several iterations (Meyer, 2014). Agile development environments are change-aware and are inherently more adaptable to evolving requirements and shifting goalposts.

In the case of the *Gibbous Earth* project, an agile approach is favoured for the following reasons:

* Game development is not regulated,
* Dynamic development environment,
* Scope to add, remove and change features during development,
* Product criticality is low,
* No legacy code replacement/integration, and
* Product is not likely to require integration with external/custom systems.

It is worth noting that Boehm and Turner (2009) describe several scenarios where teams of 50 - 150, although successful, were not easily managed. Although Saturn Games expects to employ up to 50 people, the development team is likely to consist of 20-25 employees. For this reason GWS advise an Agile-based approach.

# Crystal Development Approach

This section outlines Crystal, a development approach which meets the Agile philosophies developed by Beck, et al. (2001) such as:

1. Early and frequent delivery of working software which brings value,
2. Support for requirement-change,
3. Daily collaboration between Business & Development,
4. Support and trust of employees,
5. Maximisation of face-to-face communication,
6. Sustainable pace of development,
7. Reflection and adjustment of the development environment,
8. Efficiency through simplicity.

## Overview

The Crystal ‘family’ of methodologies are intended to provide project outlines which guide rather than dictate the requirements for an agile project. Each crystal type is identified by a colour, beginning at Clear and ranging through Yellow and Orange to Red, which indicate increased project requirements as a function of increased project size and/or criticality (Cockburn, 2007). The methodology aims to be as light as possible for any given project and requirements (properties) are tailored accordingly, with one main concept: “Osmotic Communication” (Cockburn, 2007; Meyer, 2014). Cockburn explains this is when team members work so closely they gather valuable information from conversations and work stations which occur naturally in the work environment.

All colours in the Crystal family support the following values:

1. The techniques and practices used will be people-centric and communication-centric, thereby supporting the human element (Cockburn, 2007);
2. The design environment will be highly tolerant, recognising the varied nature of human cultures (Cockburn, 2007).

These values will be essential to the *Gibbous Moon* project as the team will be newly established and will likely go through forming and storming periods before norming and performing. Crystal offers a pathway to a supportive development environment, tolerant of change and the people who implement it.

### According to Meyer (2014), small projects (teams up to eight) can utilise Crystal Clear while Orange is older and suits larger projects. Cockburn (2007) states Crystal Orange was developed for a team of up to 40 people, where the project length is expected to be one to two years and the system is not life- or legally critical. In the case of the *Gibbous Earth* project, the most suitable type of Crystal project is Yellow, which is based on an expansion of Clear and is for teams of size fifteen to 30 (Cockburn, 2007).

## Advantages

The main advantage of the Crystal methodology is that it is designed to be fine-tuned to fit the project needs. By ‘stretching’ the lighter version up for a larger team (i.e. from Clear to Yellow) Saturn Games can be sure they are not over-burdening the development team with excess managerial overhead. Regular reflection and tuning throughout the project life enables the development environment to constantly evolve, thus coping with the rapid increase in employee numbers and the forming/storming periods associated with new teams.

## Disadvantages

By Cockburn’s own admission, the lack of prescribed processes in Crystal makes it daunting for beginners in Agile development. To make the most of the approach; the development team should have strong experience in methods such as Scrum, XP, Kanban and others in order to tune Crystal to meet the needs of the *Gibbous Earth* project. This lack of prescription would require greater pre-development design, although once up and running Crystal’s tolerance of change will make it easily manageable on a day-to-day or week-to-week basis. Meyer (2014) notes the requirement to change to a new method if the size or the criticality of the project changes significantly, however; an experienced team will handle the shift smoothly, noting there is potential for some significant burden in re-tuning or changing methods for gross project changes.

## Suitability

Meyer (2014) suggests some of the practices or techniques encouraged by Crystal are overhyped, such as pair programming, sustainable pace, and open-space offices. There is no requirement to utilise these techniques if they do not suit the Saturn Games team. Once again, with a high level of Agile experience on the *Gibbous Earth* project, Crystal Clear/Yellow would provide suitable guidelines for creating an ad-hoc approach which still meets the minimum requirements for a quality product delivered from within a dynamic and safe environment.

## Practices and Requirements

The Crystal methodology does not prescribe specific techniques, instead; it provides a collection of project roles and properties which encourage a minimum overhead while ensuring completeness of project design. Cockburn (2005) instructs the following three project properties are required for Crystal Clear: ‘Frequent Delivery,’ ‘Reflective Improvement’ and ‘Osmotic Communication’ (p. ix) and GWS recommend the addition of ‘Focus’ and ‘Personal Safety’ as well. The processes and practices used to achieve these properties are chosen and defined during the initial ‘method-tuning’ stage of project design. Crystal allows for processes from other methodologies to be utilised, as long as they fulfil the requirements of the standards (Cockburn, 2007).

### Frequent Delivery

Cockburn (2005) differentiates between short iterations and frequent delivery, explaining a collection of short iterations combine to produce deliverable value to the customer (refer Fig 1).

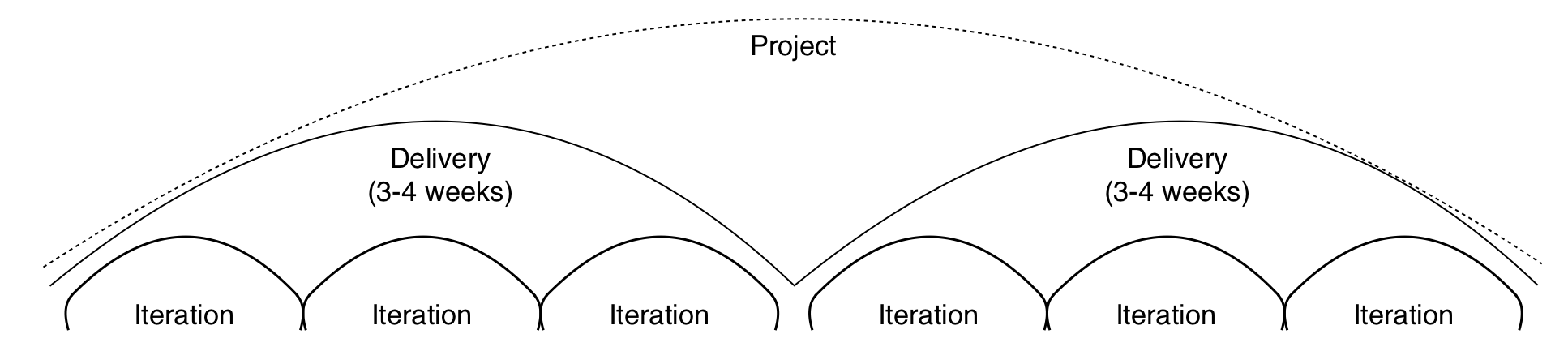


Figure 1: Short iterations and frequent delivery

(Adapted from Cockburn, Fig 4-1 (2005, p113))

Saturn Games have indicated they do not intend to deliver product which is not a fully market-ready ‘final’ version (no public alpha and presumably no beta releases), therefore; their delivery cycle is may be considered to be two years (Fig 2).

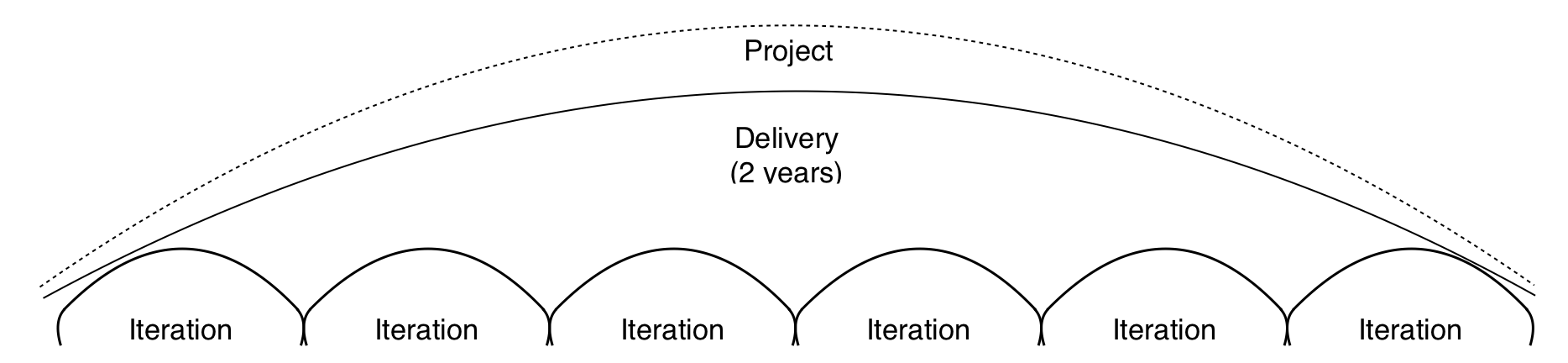


Figure 2: A single, two-year delivery

The Crystal family of methodologies, especially Clear through Orange, consider a one-year delivery cycle to be a danger sign which must be mitigated with high user involvement during development iterations (Cockburn, 2005, 2007), and the same is true for a two-year delivery. GWS makes two recommendations in this area. First, given the nature of the *Gibbous Earth* game in that a) game-play is nonlinear and b) there is room to expand and change the product concept; GWS recommend planning to include product milestones which provide definable value to the users and which can be used to determine delivery cycles. The first of these milestones is the ‘early victory’ accomplished by the development of a ‘walking skeleton,’ which test the architectural design of the product and ensures systems will integrate effectively (Cockburn, 2005), refer to the practice guide for more details on these techniques. Cockburn (2005) also advises the coordinator, business expert, lead designer and sponsor can plot delivery milestones by creating a project map and release plan during the initial planning phases. The release plan is not intended to be a static document, but will be updated after each delivery and/or iteration according to project need (Cockburn, 2005).

Second, it is recommended Saturn Games assign the role of ‘user’ to selected project members. Users will be responsible for assessing the *Gibbous Earth* product on behalf of the expected market and in this way the probability of product errors making it to release is reduced. Crystal Clear requires there to be two user viewings per release (Cockburn, 2007), which in this case can be taken to mean delivery cycle/milestone.

Short delivery cycles and shorter iterations are supporting processes for continuous integration, and Crystal requires this is also carried out regularly (Cockburn, 2005). Team members on the *Gibbous Earth* project will determine by trial and error on an iteration-by-iteration basis the most effective frequency for integration. Note that, generally speaking, shorter is better as this allows issues to surface earlier. Several times a day should be the aim, however several times a week may suffice.

### Reflective Improvement

The second of Crystal Clear’s essential properties will enable the *Gibbous Earth* development team to effectively embrace change, especially as the team grows from very few to very many. Difficulties establishing the team and processes are inevitable, therefore; the *Gibbous Earth* team need to reflect on what worked, what went wrong and what changes will be made moving forward. There are a number of techniques which can be incorporated, but; for all intents and purposes the incorporation of some form of active reflection is more important than the specific processes used (Cockburn, 2005). By taking on the mindset that every iteration serves as an experiment in the most effective way to create the product; reflection serves to fine-tune the team’s performance as well as to identify requirements for project change (big or small) and incorporate them. GWS recommends at least one reflection per iteration at the beginning of development, due to the highly dynamic nature of the growing team. The *Gibbous Earth* team may find this frequency can be lowered to once during and once after each delivery cycle as the project environment becomes more stable.

### Osmotic Communication

The advantages of such formats as co-located teams and side-by-side programming is the ability for a team member to overhear and/or contribute to conversations which they otherwise might not have (Cockburn, 2007). Cockburn also states this will ‘[lower] the cost of idea-transfer’ (p111). The *Gibbous Earth* project is not likely to be able to place all 50 team members in the same office, therefore the osmotic communication requirement of Crystal Clear is not likely to be fully achieved. Crystal Clear can be stretched to Crystal Yellow in this case, as long as team members and especially senior staff aim to encourage easy flow information wherever possible. The team should be located on the same floor, preferably in a donut shape, to reduce the maximum time to walk between any two offices. Shared spaces should be provided for team meetings and workshops.

To further support communication, Cockburn (2005, 2007) recommends the use of ‘information radiators.’ By placing project information such as plans and status, standards, goals and team agreements in large, easy-to-see locations all team members and senior staff are able to gather information on the health of the project. Meetings will ideally be held in front of these information radiators. The practice guide has more information on these radiators.

### Focus

The sudden influx of a wide range of experience levels and personalities will inevitably lead to a significant amount of meetings, workshops and training - especially at the beginning of the project. Additionally, the size of the team is much larger than a 5-8 Crystal Clear team and will be subject to a greater variety of information and communication challenges. There is the potential that an overly strong focus on process will have a negative impact on development and coding productivity. Rather than attempting to ensure meetings and other disruptions only happen during a specific part of the day, the ‘focus’ property aims to ensure they do *not* happen during specific intervals. This allows developers a set time of uninterrupted productivity (Cockburn, 2005). Cockburn recommends a two-hour time period every day, in which nothing else is scheduled and interruptions are ‘blocked.’ Once agreed on, this focus time can be communicated to all parties via an information radiator.

### Personal Safety

# Bibliography

Please refer to the bibliography in the GWS team wiki at https://autonline.aut.ac.nz.