CGHM

# Agile development model recommendation

The Company (CGHM) has been formed with the intention of producing an e-commerce website. With a team of highly skilled developers already in place, the decision needs to be taken regarding which development methodology to adapt, choosing between Extreme Programming and Scrum Methodologies. After giving a brief overview of the 2 different methodologies, their similarities and differences, I will recommend a methodology to the company that I feel best addresses their needs and delivers the greatest return on their investment.

Extreme Programming (XP) is a discipline of software development based on 5 core values, supported by principles such as economics, diversity and accepted responsibility and delivered using practices such as sitting together, pair programming, incremental design etc. that affect processes, testing and architecture. The core values are as follows:

1. Communication: essential for creating a sense of team and to ensure effective cooperation and achieved adapting practices such as sitting together and pair programming.
2. Simplicity: a system should be made simple enough to solve only today’s problems. To a highly skilled team the technologies and techniques are simple and simple solutions are easy to design, implement, test, communicate, explain and change.
3. Feedback: quick feedback is very helpful and, wherever possible, XP integrates immediate feedback into the development process by means of unit tests for each piece of code, continuous integration etc.
4. Courage: an effective team is one which should be able to take effective action in the face of fear, for example, having the courage to communicate that changes need to be made to an interface or to face negative feedback from colleagues.
5. Respect: this value should underlie all other XP values and it should be appreciated that no team member is intrinsically worth more than anyone else and the respect of developers for the customer and vice versa is of vital importance

Scrum Methodology - as opposed to traditional systems development, where the development process is well understood, can be planned, estimated and successfully completed – assumes that the development is unpredictable and complicated and therefore can only be described as an overall progression. According to the Scrum methodology, such a complex process requires empirical rather than defined process control, i.e. one based upon observation and tests as opposed to a set of rules. Some characteristics of Scrum projects are:

* Self-Organized – employees are self-motivated and seek to accept greater responsibility
* Produce Progression – development takes place in a series of month-long iterations known as sprints
* Requirements – all requirements are captured in a list known as the ‘product backlog’
* Collaboration – both within the Scrum project team members and with outside parties such as the customer, i.e. intra/inter collaboration is expected
* Frequent reviews – team progress reviewed frequently (every 1-4 weeks)

The Scrum is comprised of three distinct phases referred to as Pregame, Game and Postgame.

*Pregame* consists of:

* Planning: definition of a new release based on current, known backlog (requirements for produce being developed), along with an estimate of schedule and cost
* Architecture: design how backlog items are implemented. System architecture modification and high-level design are included in this phase

*Game* consists of:

* Development sprints: iterative cycles of development work conducted over a pre-define period of usually 2-4 weeks. Each Sprint consists of one or more teams performing:

Development – defining changes required to implement backlog items

Wrap – creating an executable version of changes and how they implement backlog items

Review – all teams meet to review progress and to assess risks

Adjustment – review information consolidated and implemented

*Postgame* consists of:

* Closure: preparation for release which includes final documentation pre-release testing and release

In the pregame and postgame phases both planning and closure consist of defined processes where all the processes and inputs/outputs are well defined and where knowledge of how to do these processes is explicit. In the game phase we have the sprints and this phase is an empirical process as many of the sprint processes themselves are unidentified and uncontrolled and so, rather than being based on a fixed set of rules, they are based on observations and tests.

Scrum implements iterative, incremental development through three roles and all management responsibilities in a project are divided. The roles are:

1. Product Owner – responsible for the success of the project and also the voice of the customer. Responsible for representing everyone with a stake in the project and its resulting product. Organises funding and creates initial product backlog. Prioritises items in backlog in terms of valuable functionality/return on investment (ROI).
2. Team – responsible for development. The team is self-managing, self-organising and cross-functional and determines how to turn backlog into a piece of functionality within an iteration. The team members are collectively responsible for the success of each iteration
3. Scrum Master – responsible for the scrum process and for teaching its philosophy, rules and practice to team members

Scrum also employs the following tools or artefacts:

* Product Backlog – this is the responsibility of the product owner and contains the requirements for the product being developed. The backlog is never complete and after the planning phase contains only the initially known and best-understood requirements, however it evolves as the product and environment emerge.
* Burndown Chart – shows the amount of work remaining and represents the reality of work completed, how fast it’s being done and what is planned and hoped for.
* Sprint Backlog – defines the tasks that the team defines for turning the product backlog selected for that sprint into an implement of potentially shippable functionality

According to Mike Cohn of mountaingoatsoftware.com there are 4 main differences, subtle but important, between XP and Scrum. The first two differences are related to iterations which typically last 1-2 weeks in XP and 2-4 weeks in Scrum, where they are called sprints. Also, Scrum teams do not allow changes into their sprints. “Once the sprint planning meeting is completed and a commitment made to delivering a set of product backlog items, that set of items remains unchanged through the end of the sprint.” Coen goes on to state that XP teams are much more open to changes being made within the iterations. For example, as long as the team hasn’t started work on a particular feature within an iteration then it can be replaced by a feature of comparable size, meaning that the iterations are much more flexible.

In XP, teams work in a strict priority order where the features to be developed are prioritized by the customer (The [Product Owner](https://www.mountaingoatsoftware.com/agile/scrum/roles/product-owner) in Scrum) and the team is required to work on them in that order. The Scrum Product Owner prioritizes the product backlog; however, it is the team that determines the sequence in which they will develop the backlog items, generally choosing to work on the most high-priority item first. So, it can be said that a Scrum team is more self-organising than an XP team.

With the Scrum methodology there are no engineering practices that the team *must* do like there are in XP (test-driven development, pair programming, simple design, refactoring etc.), rather the emphasis is on continuous improvement within the team whereby, at regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

The intention of CGHM is to produce an e-commerce website to rival Amazon and because they already have in place a team of highly experienced developers, they will expect results and ROI pretty quickly. With Scrum, a potentially shippable, functional product can be achieved relatively quickly, in theory after each sprint.

As I mentioned earlier, in XP there are set development practices to be followed by the team whereas the Scrum methodology emphasises the management side of a project and does not define a detail engineering practice. Therefore, in effect, the two methodologies are not mutually exclusive but rather can be combined and interleaved.

According to Josh Partogi from scrum.org there is no reason why a choice has to be made *between* Scrum and XP as Scrum is just a framework for product development, what he considers a container, where different practices can be added, including practices enshrined in XP.

“Adding XP into Scrum should be the next natural path for teams starting out with Scrum and striving to be a professional Scrum Team. From my experience, Scrum Team practicing XP is a hyper-performing team. XP is one of the missing piece Scrum team need to deliver great quality products. Scrum and XP are well aligned and complements each other well hence the question of whether to use Scrum or XP are irrelevant.”

So, in conclusion I feel that the best approach for CGHM to take would to adapt the Scrum development methodology while adopting XP practices such as sitting together, pair programming, test-first programming, continuous integration, incremental design etc. that have been proven to achieve speedy and effective results.

**Bibliography**

Cohn, Mike. (2007, April 6) *Differences between Scrum and Extreme Programming*. Retrieved <https://www.mountaingoatsoftware.com>

Ferolen, Admin. (2008, April 15). *XP Vs Scrum*. Retrieved from <https://www.ferolen.com/xp-vs-scrum/>

Kenirons, Martin. (2018). *Extreme Programming*. Agile Software Development Course Notes. Retrieved from <https://learnonline.gmit.ie>

Kenirons, Martin. (2018). Scrum. Agile Software Development Course Notes. Retrieved from <https://learnonline.gmit.ie>

Partogi, Joshua. (2017, March 17). Scrum And eXtreme Programming (XP). Retrieved from <https://www.scrum.org/resources/blog/scrum-and-extreme-programming-xp>