Word Count :1606 exclusive of tables

Introduction

This report will examine the Agile methodologies Scrum and Extreme Programming in relation to the development of a new e-commerce website CGHM that is poised to compete with e-commerce website Amazon.

The author will research the aforementioned Agile Methodologies by consulting research papers and case studies. The author will then conclude with the chosen methodology, outlining reasons for that choice.

The Competitor

Amazon, an enterprise which has branched out to many industries since its creation in 1995 as a book retailer. For the purpose of this report, the author will focus on the an e-commerce website.

Development of Amazon

Currently Amazon uses the Scrum methodology in its development, using the AWS platform. Amazon is seen as a role model in terms of the UX design of the site, with University of Michigan using it as a positive example of how to apply UX principles.

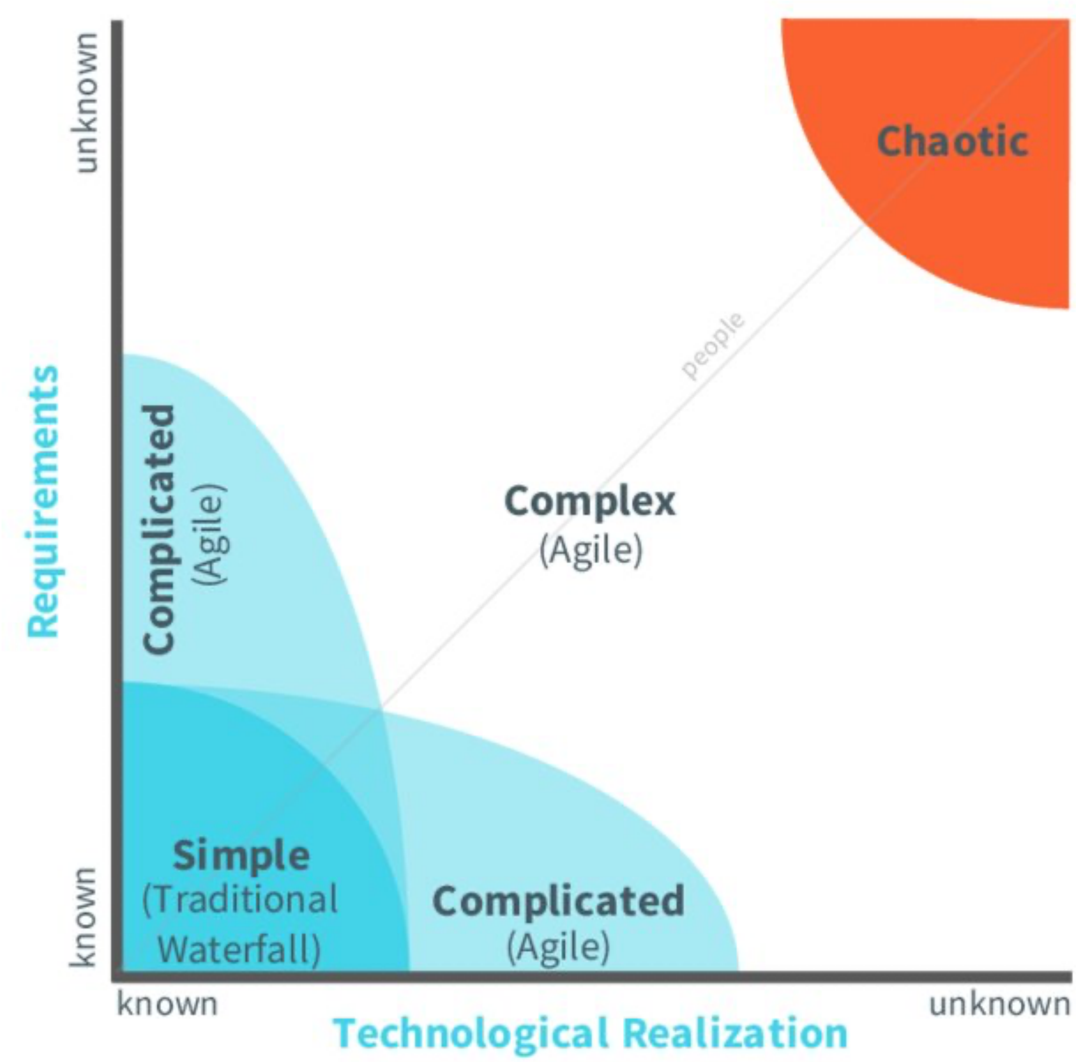
The Product

The product that is proposed for development is an e-commerce website entitled CGHM. The existing site layout and services of Amazon can form a blueprint for the design of CGHM.

In order to assess which methodology will need to be used, the author will ascertain the complexity of the project: CGHM is a large scale project, requiring a large development team and will be required to work with using third parties which can create variables within development. Therefore the development process of this project is Complex.

Complexity of Software Development Process	
Simple	Everything is known
Complicated	More is known that unknown
Complex	More is unknown that known
Chaotic	Very little is known

Working as scrum master in an agile team, p.27



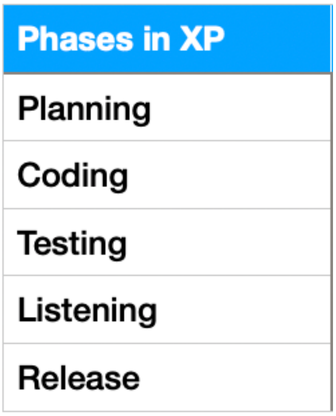
Graph showing how complexity is affected based on amount that is known vs unknown Complexity and Organisational Reality', D. Stacey

Extreme Programming

Extreme programming (XP) began in 1996 with Ken Beck ET all, with the aim of avoiding lengthy cycles common with traditional development cycles, as well as producing higher quality software and a higher quality of life for the development team. (What is Extreme Programming (XP), 2020)

In relation to appropriate engineering practices XP is the most specific of existing agile methodologies. XP is most appropriate in small co-located development team, for small scale projects where there are dynamically changing software requirements, and its structure responds well to a changing environment. XP, which assumes chaos, encourages short term planning which reduces chaos, whilst being highly adaptive toward requirement changes and consists of highly productive teams that produce quality software rapidly and effectively. (Panayotova, 2020) Xp encourages customer involvement, as well as prioritising testing, high quality engineering, feedback, planning and collaborative teamwork.

During the planning phase, the customer, developers and testers specify requirements prior to project commencement, thus creating a specification from the data obtained within the User Stories. The development team delivers tested working software at the end of each iteration, with initial versions of the system with a simple design released early in the process, progressing to releases of more detailed versions as the development progresses.



Phases in XP
Planning
Coding
Testing
Listening
Release

Rapid, Fine Feedback	
Testing	- continual tests result in a system that can accept change
On-site customer	- team user available full time to deal with queries and issues
Pair programming	- code written by two developers sharing one device for production of high-quality code
Continuous Process	
Continuous Integration	- code is integrated and tested several times a day
Refactoring	- restructure system to simplify, add flexibility, remove duplication or improve communication
Short Releases	- releases of working systems containing most valuable and immediate requirements
Shared Understanding	
The Planning Game	- implementation of plans prior to production, create blueprint to reduce chaos
Simple Design	- simplest design possible to be used at any given moment
Metaphor	- simple shared story of how the entire system architecture operates
Collective Ownership	- code is available to all, beneficial when code needs to be modified.
Coding Standards	- all developers create code in accordance with the rules in order to allow adaptation within the team and consistent development
Developer Welfare	
Forty-Hour Week	- adhering to this Principle allows staff to be rested.

Table showing Extreme Programming Practices , grouped into sections. Adapted from (Extreme Programming - Practices - Tutorialspoint, 2020)

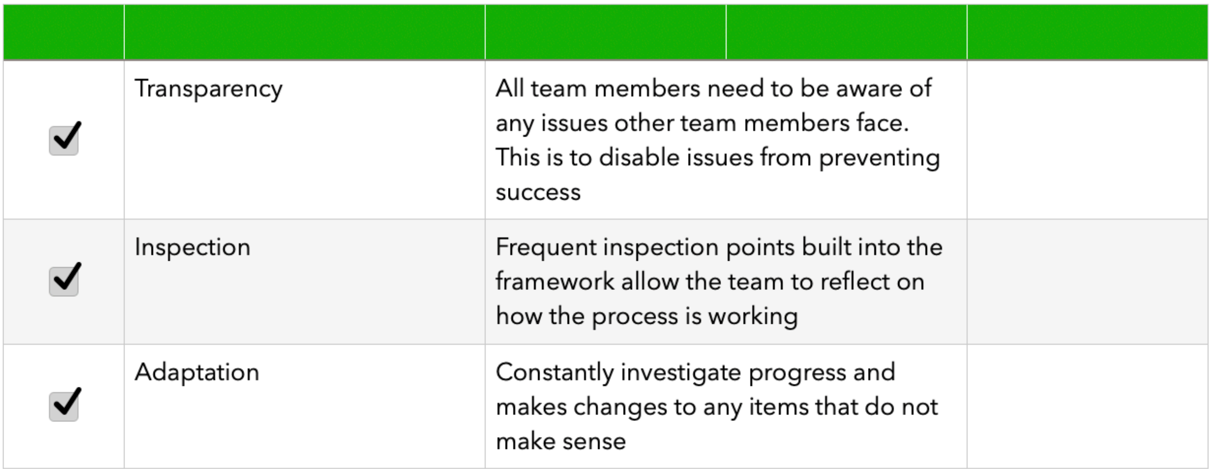
Scrum

Scrum, an agile project management framework, was first implemented in 1993. “Scrum is a framework for developing, delivering and sustaining complex products” (Working as a Scrum Master in an Agile Team, Delegate Guide) Scrum is orientated around best practice project management techniques, describing a set of principles to guide teams in communication, tools, and team roles that assimilate to enable self organised teams to structure and manage their work.

Scrum employs an iterative (repeated cycles), incremental(small portions at a time) approach to optimise predictability and control risk. (Scrum, 2020)

5 Scrum Values		
✓	Courage	Do the right thing and work on tough problems
✓	Focus	Concentrate on the work identified for the sprint and the goals of the team
✓	Commitment	Team members personally committed to achieving team goals
✓	Respect	Team members respect each other to be capable and independent
✓	Openness	Team members and stakeholders are open about all work and challenges the team encounters

Table depicting 5 values of scrum, adapted from Working as a scrum master in an agile team, Delegate Guide



☑	Transparency	All team members need to be aware of any issues other team members face. This is to disable issues from preventing success	
☑	Inspection	Frequent inspection points built into the framework allow the team to reflect on how the process is working	
☑	Adaptation	Constantly investigate progress and makes changes to any items that do not make sense	

3 Pillars of Empirical Process Theory, (Working as a scrum master in an agile team, Delegate Guide)

Scrum is founded on empirical process control theory. As Scrum teams are self organised, these principles allow teams to establish a hypothesis of how they think something works, try it out, reflect on the experience and make appropriate changes.

> *"Self organising teams just do a better job no matter what they do" ; Jeff Sutherland*

The three roles in teams :

Scrum Master

- in charge of modifying code for validation at the end of the process.
- solves problems that stop the team from working effectively.
- ensure team members understand goals, scope, and product domain.
- leads and coaches the organisation in its scrum adoption
- promotes and supports scrum as defined in the scrum guide
- serves development teams, product owners and the wider development organisation
- coaches development teams
- facilitates many events and conversations

Product owner

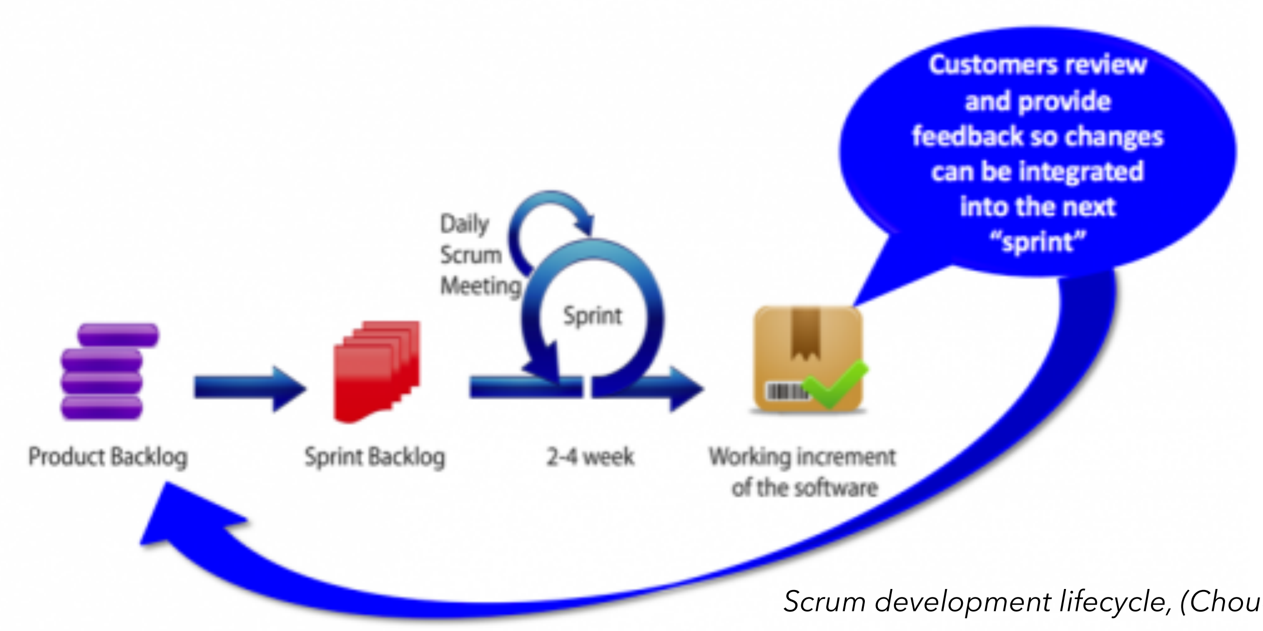
- ensures the backlog is visible, transparent and clear to all
- adds items to and orders product backlog
- manages relationships with stakeholders to ensure good relationship management
- engage with development teams to refine product backlog
- remain accountable even if tasks are delegated.

Development team (5-9 members) carry out work, complete potentially releasable product at the end of each sprint. Self organised and empowered, manage their own work. Accountable and committed; accountability belongs to the development team as a whole. Can include Programmers, Software developers, Testers, QA, UX Designers, Graphic Designers, Business Analysts, Copywriters, Technical Architects.

	Three Product Artefacts of Scrum	
✓	Product backlog	High level list maintained throughout the product life cycle
✓	Sprint backlog	List of PBIs development team selected for the sprint
✓	Product increment	Value of potentially releasable work completed for sprints that meets common and transparent definition of done.

Software is developed in increments called sprints , beginning with planning and concluding with a review. Features to be implemented in the system are registered in a backlog.

	Five Events of Scrum
✓	Sprint
✓	Sprint Planning
✓	Daily Scrum
✓	Review
✓	Retrospective

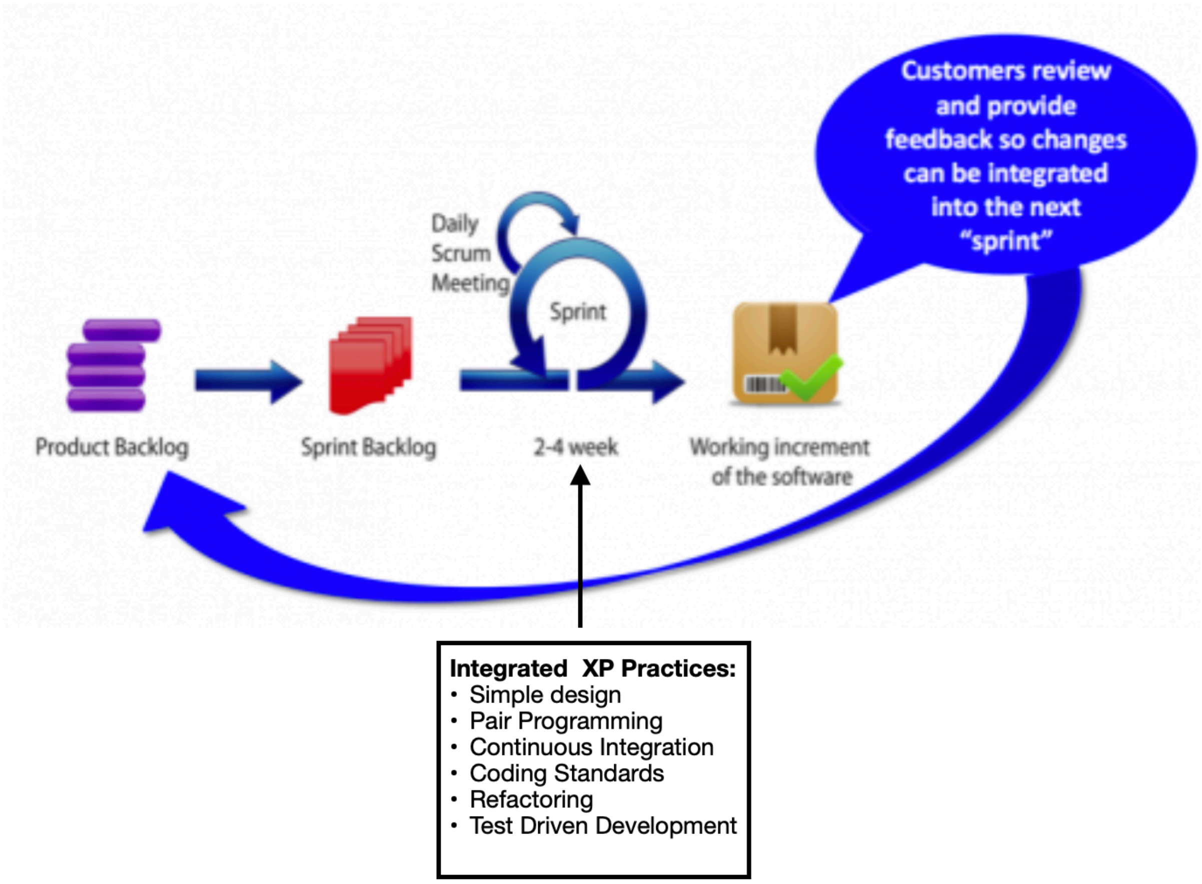

Customers review and provide feedback so changes can be integrated into the next "sprint"
Daily Scrum Meeting
Sprint
Product Backlog
Sprint Backlog
2-4 week
Working increment of the software

Scrum development lifecycle, (Choudry, 2020)

Proposal:

Both Scrum and XP have valuable features whilst having their own limitations. XP has sound engineering practices, yet is not suited to large scale projects, whereas Scrum is well matched to large scale projects due to detailed management practices. The best practice approach for CGHM would be a novel hybrid model that uses Scrum's management framework as the umbrella methodology along with selected engineering practices from XP's framework thus removing any limitations in each methodology.

Importantly, Scrum is designed to be flexible and is designed in such a way that teams have the ability to incorporate in beneficial practices from other Agile frameworks. Many teams start using a different framework and when they identify the need for more disciplined engineering practices they cherry-pick some or all of the engineering practices detailed by XP. (Novel Hybrid Model: Integrating Scrum and XP ,2012)


Customers review and provide feedback so changes can be integrated into the next "sprint"
Daily Scrum Meeting
Sprint
Product Backlog
Sprint Backlog
2-4 week
Working increment of the software

Integrated XP Practices:
- Simple design
- Pair Programming
- Continuous Integration
- Coding Standards
- Refactoring
- Test Driven Development

Proposal : Scrum development life cycle, adapted to include 6 XP Practices, diagram adapted from (Choudry, 2020)

Scrum Sprints are an ideal element within certain elements of XP can be introduced. Integrating Xp's testing and code-refactoring into the Sprint would be beneficial to the project by adding value overall in ensuring high quality tested code is created. For the purpose of this report, the author created a table checklist in which relevant XP practises were selected. (appendices)

The Sprint Stage to include the XP practices :

Simple design: As in XP, keep design simple by only coding what is necessary at the given stage of the project.

Test Driven Development: At the beginning of the Sprint, test driven development should occur. Clutterbuck, Rowlands & Seamons, found this resulted in timelier, more accurate design changes with fewer design changes, as well as an alignment of business functionality and software quality.

Refactoring : Code refactoring should occur at regular intervals. Using a simple design with along with scheduled testing. The IXSCRUM case study found that their model of combining testing using HP Quick Test tool and a Scrum management too worked well, enhancing the quality of both Scrum and Xp, reducing any drawbacks. (IXSCRUM Framework Combining Scrum and XP,2012)

Pair programming (flexible): to be introduced as in FinApp case study (How extreme does Extreme Programming have to be? Adapting XP practices to large-scale projects., 2004) The FinApp case study reported that programmers found it useful to run test case development, unit testing , design, analysis in paired formation, and took the option to code solo. The benefits of this approach include: Reduce Development time, Reduce Training time, Improve Quality of Code, Create a Collaborative and supportive environment.

Collective ownership: code is available to all, beneficial when code needs to be modified, increased adherence to software architecture standards, promotes teamwork, greater knowledge of software architecture.

Continuous integration: code is integrated and tested several times a day, increased flexibility in personnel management, increased capability for testing.

Coding standards: all developers create code in accordance with the rules in order to allow adaptation and readability within the team and consistent development

XP practices not included are:

Planning game: Clutterbuck, Rowlands & Seamons, states there was exceptional focus on personal or technical issues and not enough on business issues. (A Case Study of SME Web Application Development Effectiveness via Agile Methods)

Forty-hour week: there exists an argument of adopting XP’s 40 hour week to ensure rest, although all interviewees in the FinApp case study found this unworkable

Short release cycles: Large projects, such as FinApp, the Architectural Design took 6 months. It has to be imagined that CHGM, a large scale project, would require an extended architectural design stage. This could then be followed by short iterative cycles.

Metaphor : For complex large scale projects design in advance is considered essential, therefore XP’s metaphor will not be used. The case study for FinApp, the researchers found that designing up front was necessary, which is the case here as we have a blueprint existing of what the website needs to be structured like.

On-site customer: Customer on site during planning, pre sprint and closure (customer sign off on deliverables)

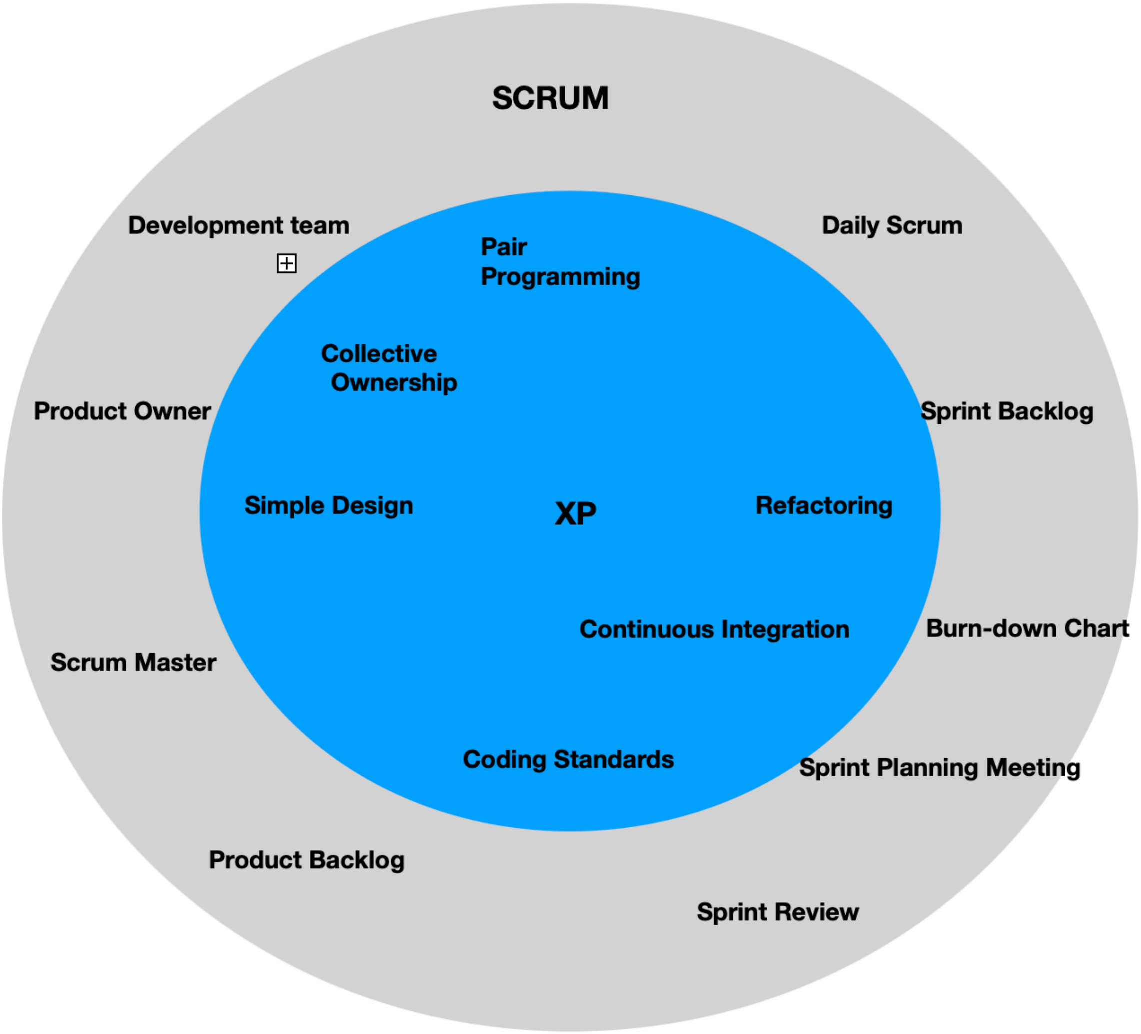


Diagram detailing the proposed integration of 6 XP practices into Scrum, adapted from Getting Agile Methods to Work for Cordys Global Software Product Development (2011)