

Custom Software Solutions within the “Managed Agility” framework



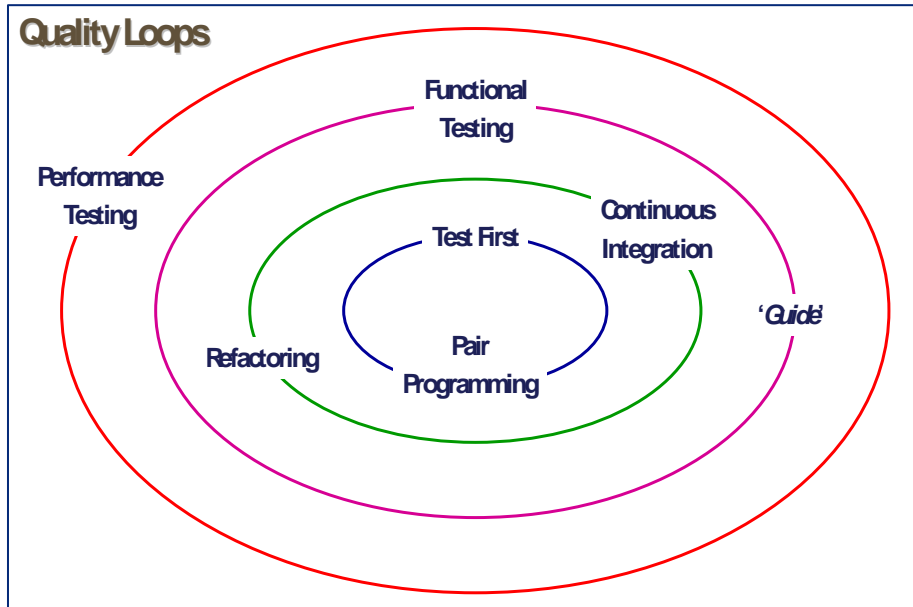
Over the past years, 4C Technologies has thoroughly investigated in implementing software development projects using iterative-incremental principles.

Upon the great results 4C Technologies has developed a custom framework, called “Managed Agility”. This framework holds elements from Scrum and eXtreme Programming, worldwide the most adopted ‘Agile’ methods.



The base of the framework is [3R4Q] i.e. how the definition of 3 R's can lead to Q (for Quality):

- A number of roles –R1– (see further).
- Description of Requirements –R2– in essential Use Cases and User Stories.
- Splitting of Releases –R3– within the project elapse time in a number of iterations, over which functionalities is incrementally built. In between the iterations concise feedback points are foreseen for synchronization of expectations and priorities.
- An active, integrated quality policy –Q–, according to following 4 ‘Quality Loops’ (*see picture next page*).



Roles in the Managed Agility framework

4C Technologies foresees following roles on a project team within an iterative-incremental ('agile') approach:

- A **Scrum Master** is responsible for mastering the whole of a project and the project's process, maintaining and promoting the project's vision and the practices and principles. The Scrum Master uses empirical principles and face-to-face communication over predictive tools. The scrum master applies tracking methods from the 'managed Agility' framework to determine adaptive measurements and install these measurements in close cooperation with the team.
- A **Coach** assists the project team for technical guidance and steering. The Coach actively participates in (pair) programming. The Coach maintains and promotes the design philosophy, and guards correct application of the technical practices from the Quality Loops.

A **Product Owner** assists the project team for functional details and steering. The Product Owner actively manages the project's Product Backlog. This includes prioritizations and close collaboration with the customer/user. The Product Owner shares the tracking responsibility with the Scrum Master.

- A **Guide** generates a continuous flow of functional feedback to the project team through daily testing. The Guide produces functional test scenarios that are actively verified with the customer/user.
- A **Developer** frequently picks and develops User Stories (essential Use Cases). Development is done using test-based development and pair programming. Code is regularly checked-in in a system of continuous integration. A developer uses refactoring techniques in order to improve the overall result. The developer works with the Coach on incremental design.
- An **MDI** team ('Multi-Disciplinary Involvement') assists the project team in specific tasks like GUI-design, system engineering, DB management and administration, performance, etc.

These are all value adding roles and are thus required to be full-time available on a project. This holds that these project members are committed to and have direct influence on the project result and quality.

This is opposed to the (past) industrial idea of separated, non-value adding responsibilities like designers, supervisors, managers, (quality) controllers etc., instructing the 'workers' what to do.

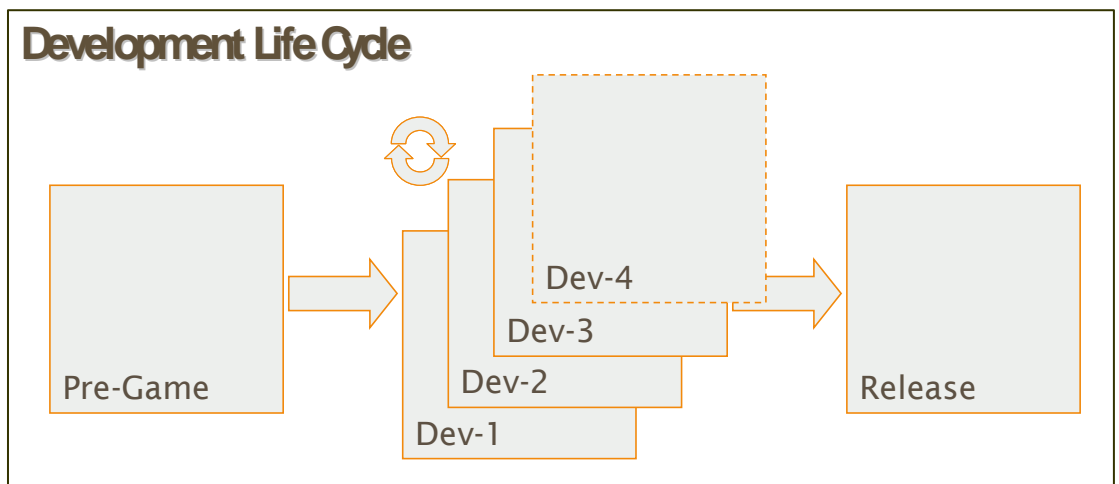
A project team thus becomes an expert group that should be self-challenging and

Profile	Project Role
Project Management	Scrum Master
Functional Analyst	Product Owner
Software Architect	Coach
Application Tester	Guide
Development	Developer
Database Designer, DBA, System Engineer, Senior Software Architect	MDI

Following table maps traditional roles on 'Lean' roles:

Iterative approach

Over the elapse time of a project, 4C determines a number of iterations according to following development life cycle:



4C suggests organizing following phases in the given timeframe:

Sprint 0

Sprint 0 holds a pre-game staging phase (also: Inception). The timeframe for Sprint holds:

- Organization of workshops to finalize the content of the first release (Product Backlog) and further refine the development requirements for the first Sprint (Sprint Backlog).
- Setup of a development environment at 4C:
 - 4C will supply the development PC's.
 - The customer is required to have an acceptance and production infrastructure (hardware device) ready before the first Factory Acceptance Test.

Sprint 1-n

Sprints 1-n cover development and are known as the Project Services.

In parallel with the development, test cases are being defined and integrated in scenarios.

Customer acceptance will take place for every Sprint after internal tests by 4C and transfer of the test results in writing to <client>. This is considered a FAT ('Factory Acceptance Test').

Sprint m

This Sprint covers installation, release and training activities.

4C will transfer all custom developed software to the customer. Customer acceptance will take place after transfer of the equipment to the production environment, application tests by 4C and transfer of the test results in writing to the customer. This is considered a SAT ('Site Acceptance Test').

Documentation and deliverables

4C will make up technical and API documentation on the implemented system. The documentation will be delivered during the Release Sprint.

At the same time 4C will organize an operator training.