JHONATHAN AUGUSTO DUARTE DE CAMPOS
SE 4TH GEN
PM_07 Agile Process Management
Project link: https://morning-forest-34963.herokuapp.com/

The module requirements will be explored amidst the content of the Family Tree Project, a digital service that aims to enable family members to share memories (videos, photos, essays) through generations on a reliable, easy-to-use, long-standing, and privacy compliance platform. In the future, when possible, we aim to enable DNA storage, allowing many generations of a family to access the data if previously allowed by the original data owner.

The project implementation started at CODE 1st semester with the problem definition going through product idealization until the early design mock-ups and the user storyboards. With the project idea developed and explained and with the primary user boards ready, we had to decide how to implement the software development. A research/study phase was necessary.

At first, knowledge about Agile and its frameworks was acquired with Google Agile Course on Coursera, with the Agile Learning Unit at CODE, and with the books Clean Code: A Handbook of Agile Software Craftsmanship and Clean Agile: Back to Basics by Robert C. Martin, Scrum: The Art of Doing Twice the Work in Half the Time by Jeff Sutherland and the Scrum Guide from Scrum.org web site.

With the key concepts in mind, the team had to consider the available methods available and mostly known: Waterfall or Agile.

The waterfall is an early development model that surfaced in early 1970, and it is a streamlined approach to building a product.

It has the following phases: requirements specification, design, implementation, verification, maintenance, and a strong emphasis on the planning and specification parts, with a firm order of the project phases – a new phase is not allowed before the end of the previous phase.

It works very well for defined and single deliverable software because it demands, as stated, extensive planning and documentation and control for low-risk and predictable products.

Software development is complex, as we know, and if one must wait for the software to be fully completed before release, the cost, quality, and efficacy would be extremely high. That is when the Agile mindset comes in.

Unlike the waterfall traditional method, the Agile methodology was introduced with the Agile Manifesto, in early 2000, aiming to implement and deliver a project quickly. It allows the team to start the development phase without long planning and also to deliver incremental working functionality fast.

The Agile principles were essential to direct our actions and states that it is our highest priority is to satisfy the customer through early and continuous delivery of valuable software; the highest priority is to satisfy the customer through early and continuous delivery of valuable software; deliver working software frequently; business people and developers must work together daily; build projects around motivated individuals; prioritize face-to-face conversation; working software is the primary measure of progress; agile processes promote sustainable development; the sponsors, developers, and users should be able to maintain a constant pace indefinitely; continuous attention to technical excellence and good design enhances agility; the best architectures, requirements, and designs emerge from self-organizing teams, and at regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

It was clear that Agile development would require enthusiasm for communication, transparency, and openness to quickly respond to change, to have customer collaboration over contract negotiations, and to find new ways to improve and to learn from mistakes.

For us, the perfect framework that allowed us to work the agile principles is Scrum with the addition of the Kanban Board and Test-Driven Development, which was introduced with the Extreme Programming methodology.

From the Kanban Framework, we used the board to help visualize work, maximize the flow and to limit work-in-progress. We decided to use the Kanban board instead of the Scrum Board since the first mentioned is more fluid and more easily adapted, while the latter follows a rigid methodology.

The choice for Scrum was clear because it is a framework for getting things done with the agile mindset. By using Scrum, we would have a basic flexible structure to start living the agile principles, which would allow the team to learn through experiences, self-organize while working on a problem, and reflect on their rights and wrongs to improve continuously.

It is structured to help the whole team to adapt to the change of conditions and/or user requirements, with re-prioritization built into the process and the short iteration that allows for fast release cycles, with constant learning and improvement.

The roles in the Scrum Framework are Product Owner, Scrum Master, and the Development Team.

The first one – Product Owner - ensures that the team will deliver the most value and tells the development what is essential to deliver (manage scrum backlog, release, and stakeholder management).

The Scrum Master, in sum and simply put, helps the Product Owner to define values and the development team to deliver them while taking responsibility for blockers that might mine the team productivity. It is a supportive and descriptive role focusing on transparency, empiricism, self-organization, and values.

The Development is a team of technical roles that creates and implements the software itself.

There are some Scrum artifacts – information needed for development - attributed to some team members that are essential: Product Backlog, Sprint backlog, and increments. They are essential tools because they provide the attributes of inspection, adaption, and transparency.

The product backlog is a list of features, tasks, requirements, and enhancements initially built in the orientation semester, and it was updated on-demand as new relevant information arrived.

The sprint backlog comprises tasks to be implemented/developed in the following product increment. The tasks are retrieved from the product backlog and divided into small sprint items.

The product increment, for its turn, is the product deliverable produced in the Sprint.

With the roles in mind, every team member had to learn about the critical scrum activities. They are Project Planning, Release Planning, Sprint, Sprint Planning, Daily Scrum, Sprint Review, and Retrospective.

So far, at this phase of the development, we aimed to create the main webpage, with sign up, sign in, user page, change password, confirmation e-mail, connection to a database, database planning, information page, and Technical Writing. Some key features are left out to other semesters because they demand knowledge that has not been studied yet. The memories visualization three will be built in React, which will be added at a later date.

The project/release planning was ready since the end of the first semester and, by definition, are, respectively, the decision about the product vision and product roadmap and about what set of product features will be released and when. We had the user storyboard, and the release plan was to end the current semester with the features mentioned above completed.

The next step was, then, to decide what product feature would be implemented on every Sprint. Sprint is a short development cycle to create a product functionality ready to be delivered. They are also called iterations and can typically last from one to four weeks. It is planned on a meeting where the team commits to a sprint goal, dividing the individual tasks that will be completed on each requirement.

During the Sprint phase, a daily short scrum meeting was held to decide the work priorities, to share what was accomplished the day before, and also to share any block.

After the iteration, a Sprint review meeting introduced by the project owner allowed the team to demonstrate the working product functionality, followed by the Sprint Retrospective to address any raised issue and what could change for the next iteration.

The first Sprint was to create the Flask Framework structure, with the main page with navigation links and footer as templates for all the other pages.

The second Sprint was to create the HTML and CSS elements of the page, focusing on the main page design.

The third Sprint goal was to create a database connection and test its functionality. We used SQL Alchemy due to its easy integration with Flask.

The fourth Sprint was longer and aimed to create the sign-up, sign-in process, including the e-mail connection to send confirmation messages for registration and password changes.

The fifth Sprint goal was to study and write the technical documentation.

The last Sprint goal was to set the continuous delivery and Integration (CD/CI) of the software from the local development drive to GitHub and, then, to the Heroku server.

Every Sprint allowed the team to focus on small deliverable increments and provided time to learn and search for help since it is all new and we are all in the learning phase. The step-by-step planned Sprint allowed us to safely create a Minimum Viable Product – MVC without the hassle of being lost about where to start and where to finish. It was all previously set on the product and release planning, which was also another great Agile advantage: to have the development road previously set and known.

Some complex tasks to mention were to divide the roles of Product Owner and Scrum Master in a small team and to define the sprint length, especially the first ones, due to the lack of knowledge about time and resources necessary. Also, the lack of previous documentation might be a weak point for the developers in complex projects.

In sum, the methodology and frameworks provided a secure path to develop the product so far and better the team integration and openness to evolve further. It is sure a great resource to be implemented in every project.