# History of Extreme Programming


- XP's origin dates back to the 90s when Kent Beck created it, when hired to lead Chrysler's Comprehensive Compensation System team
- Project had started in 1993 and by 1996 hadn't progressed much
- In XP, Practices such as pair programming and TDD were applied with great success
- Ron Jeffries, a friend of Beck was brought in to coach C3's team
- In 1999, Kent Beck formalized the practices, principles, and values of XP

# What is XP?

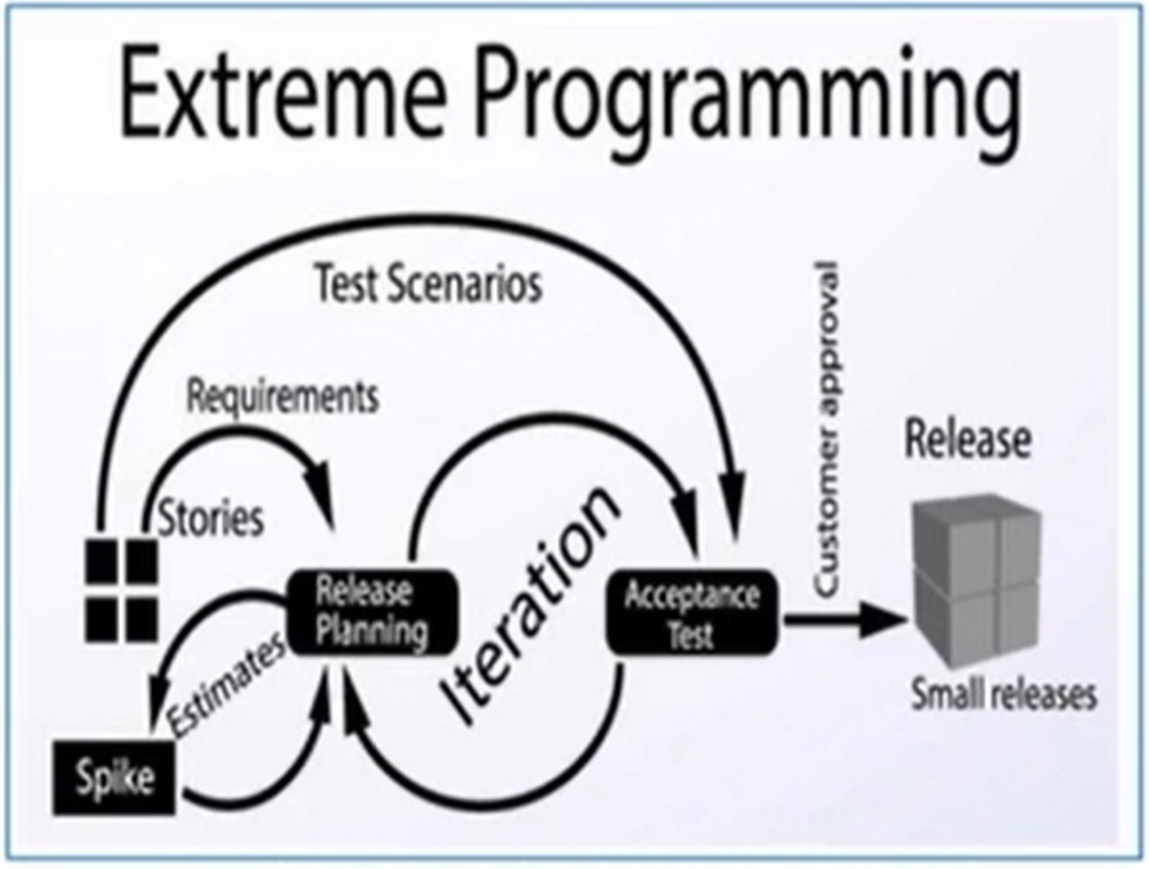
TEKNOTURF

- Definition
  - Extreme Programming(XP) is a light weight methodology for small to medium sized teams developing software in the face of vague or rapidly changing requirements

# What Is XP?

TEKNOTURF

- XP stands for Extreme Programming
- It is a software development methodology that is part of agile methodologies
- It is built upon values, principles and practices
- It emphasizes teamwork
- It allows small to mid-sized teams to produce high-quality software
- It empowers developers to confidently respond to changing customer requirements even late in the cycle
- It stresses customer satisfaction
- It also emphasizes the technical aspects of software development




https://www.digite.com/agile/extreme-programming-xp/

# Why Extreme Programming?

TEKNOTURF

- It works towards providing iterative and recurrent software releases throughout the project
- These short iterative cycles help both team members and customers to assess and review the project's progress throughout its development

# Advantages Of Extreme Programming

- Fast
- Visible
- Reduces costs
- Teamwork

# When To Use Extreme Programming?

- Highly-adaptive development
- Risky projects
- Small teams
- Automated testing
- Readiness to accept new culture and knowledge
- Customer participation

# Process Of Extreme Programming

- Planning
- Designing
- Coding
- Testing
- Listening

# Process Of Extreme Programming(Continued …)

TEKNOTURF

- Planning
  - It is the first stage
  - Customer meets the development team and presents the requirements in the form of user stories
  - Team then estimates the stories and creates a release plan broken down into iterations
  - Programming team prepares the plan, time, and costs of carrying out the iterations
  - Individual developers sign up for iterations
- Designing
  - It is a part of the planning process
  - A good design brings logic and structure to the system
  - It is related to one of the main XP values, simplicity
  - Using systems metaphor or standards on names, class names and methods to ensure compatibility
  - Using Software Class Responsibilities and Collaboration (CRC) cards to contribute ideas, and collate the best ideas into the design
  - Creating spike solutions or simple programs that explore potential solutions for a specific problem

# Process Of Extreme Programming(Continued …)

TEKNOTURF

- Planning
  - It is the first stage
  - Customer meets the development team and presents the requirements in the form of user stories
  - Team then estimates the stories and creates a release plan broken down into iterations
  - Programming team prepares the plan, time, and costs of carrying out the iterations
  - Individual developers sign up for iterations
- Designing
  - It is a part of the planning process
  - A good design brings logic and structure to the system
  - It is related to one of the main XP values, simplicity
  - Using systems metaphor or standards on names, class names and methods to ensure compatibility
  - Using Software Class Responsibilities and Collaboration (CRC) cards to contribute ideas, and collate the best ideas into the design
  - Creating spike solutions or simple programs that explore potential solutions for a specific problem

# Process Of Extreme Programming (Continued

- Coding
  - It is the phase during which the actual code is created by implementing specific XP practices
  - XP Programming gives priority to the actual coding over all other tasks
  - Standards related to Coding
    - Developing the code based on the agreed metaphors and standards
    - Pair programming aimed at producing higher quality code
    - Frequent integration of the code to the dedicated repository
- Testing
  - It is the core of extreme programming
  - It is a regular activity that involves both unit tests and Acceptance tests
  - Extreme program integrates testing with the development phase
  - All codes have unit tests to eliminate bugs
  - Acceptance test run at the completion of the coding

# Process Of Extreme Programming (Continued ...)

TEKNOTURF

- Listening
  - It is all about constant communication and feedback
  - Listening to the client is very important to see what they expect the system to do
  - Basis of feedback is Acceptance tests
  - Apart from the customer, the developer also receives feedback from the project manager
  - Each feedback of the customer becomes the basis of a new design, and the process of design-coding-tests-listening repeats itself

# XP Values

TEKNOTURF

- Extreme Programming (XP) incorporates the following five values:
  - Communication
  - Simplicity
  - Feedback
  - Respect
  - Courage


https://www.digite.com/agile/extreme-programming-xp/

# XP Practices

- Planning Game
- Small Releases
- Metaphor
- Simple Design
- Testing
- Refactoring
- Pair Programming
- Collective Ownership
- Continuous Integration
- 40-hour week
- On-site Customer
- Coding Standard

# XP Practices (Continued …)

TEKNOTURF

- Planning game
  - This is a meeting that occurs at the beginning of an iteration cycle
  - At the end of the planning game, developers plan for the upcoming iteration and release
- Small releases
  - Small releases allow developers to frequently receive feedback, detect bugs early, and monitor how the product works in production
- Metaphor
  - Metaphor stands for a simple design that has a set of certain qualities
  - First, the design and its structure must be understandable to new people
  - Second, the naming of classes and methods should be coherent

# XP Practices (Continued …)

TEKNOTURF

- Simple Design
  - Best design for software is the simplest one that works
  - If any complexity is found, it should be removed
  - Best approach is to create code only for the features you are implementing
- Testing
  - Developers continually write unit tests, which need to pass for the development to continue
  - Customers write tests to verify that the features are implemented
  - Tests are automated
- Refactoring
  - To deliver business value with well-designed software in every short iteration, XP teams also use refactoring
  - Refactoring is about removing redundancy, eliminating unnecessary functions and increasing code coherency

# XP Practices (Continued …)

- Pair Programming
  - This practice requires two programmers to work jointly on the same code
  - While the first developer focuses on writing, the other one reviews code and suggests improvements
- Collective Ownership
  - This practice declares a whole team's responsibility for the design of a system
  - It encourages the team to cooperate more and feel free to bring new ideas
- Continuous Integration
  - Code is integrated and tested many times a day, one set of changes at a time
  - Developers always keep the system fully integrated

# XP Practices (Continued …)

TEKNOTURF

- 40-hour week
  - XP projects require developers to work fast, be efficient, and sustain the product's quality
  - In XP, the optimal number of work hours must not exceed 45 hours a week
  - One overtime, a week is possible
- On-site Customer
  - End customer should fully participate in the development
- Coding Standard
  - A team must have common sets of coding practices, using the same formats and styles for code writing
  - Code written according to the same rules encourages collective ownership

# Roles And Responsibilities

- Customer
- Developer
- Manager
- Coach
- Tracker

# Roles And Responsibilities (Continued …)

- Customer
  - Customer's role is as crucial as the developer's role
  - Expected to actively participate in the project
  - Customer makes all the business decisions regarding the project
  - Communicates with the team and speaks as a single voice to the team
  - Customer could be multiple stakeholders or a community
- Responsibilities of Customer
  - Writing user stories
  - Writing functional tests
  - Setting priorities on the stories

# Roles And Responsibilities (Continued …)

- Developer
  - Developer  is the one actually creating the product
  - Realizes the stories identified by the customer
  - Knows what is needed, with clear declarations of priority
  - Developer can make/update the estimates
  - Sometimes developer can be a Programmer, Tester, Designer, Interface Designer or Network Designer
- Responsibilities of Developer
  - Estimating tasks/stories
  - Defining tasks from stories
  - Writing unit tests
  - Refactoring
  - Writing code to pass the written unit tests

# Roles And Responsibilities (Continued …)

TEKNOTURF

- Manager
  - This role falls somewhere in the middle between Coach and Tracker
  - Manager drives the strategic vision for the project
  - Ensures that the project is delivered
  - Monitors the planning game, fixes deviations, modifies rules as and when required
  - Schedules and conducts the Release planning and iteration planning meetings
  - Manager has to ensure whether the team can work towards the next release
- Responsibilities of Manager
  - Tracking the defects of functional tests
  - Tracking the time spent by each team member
  - Providing feedback at the team level and at the individual level on the estimates that helps in coming up with better estimates next time

# Roles And Responsibilities (Continued …)

TEKNOTURF

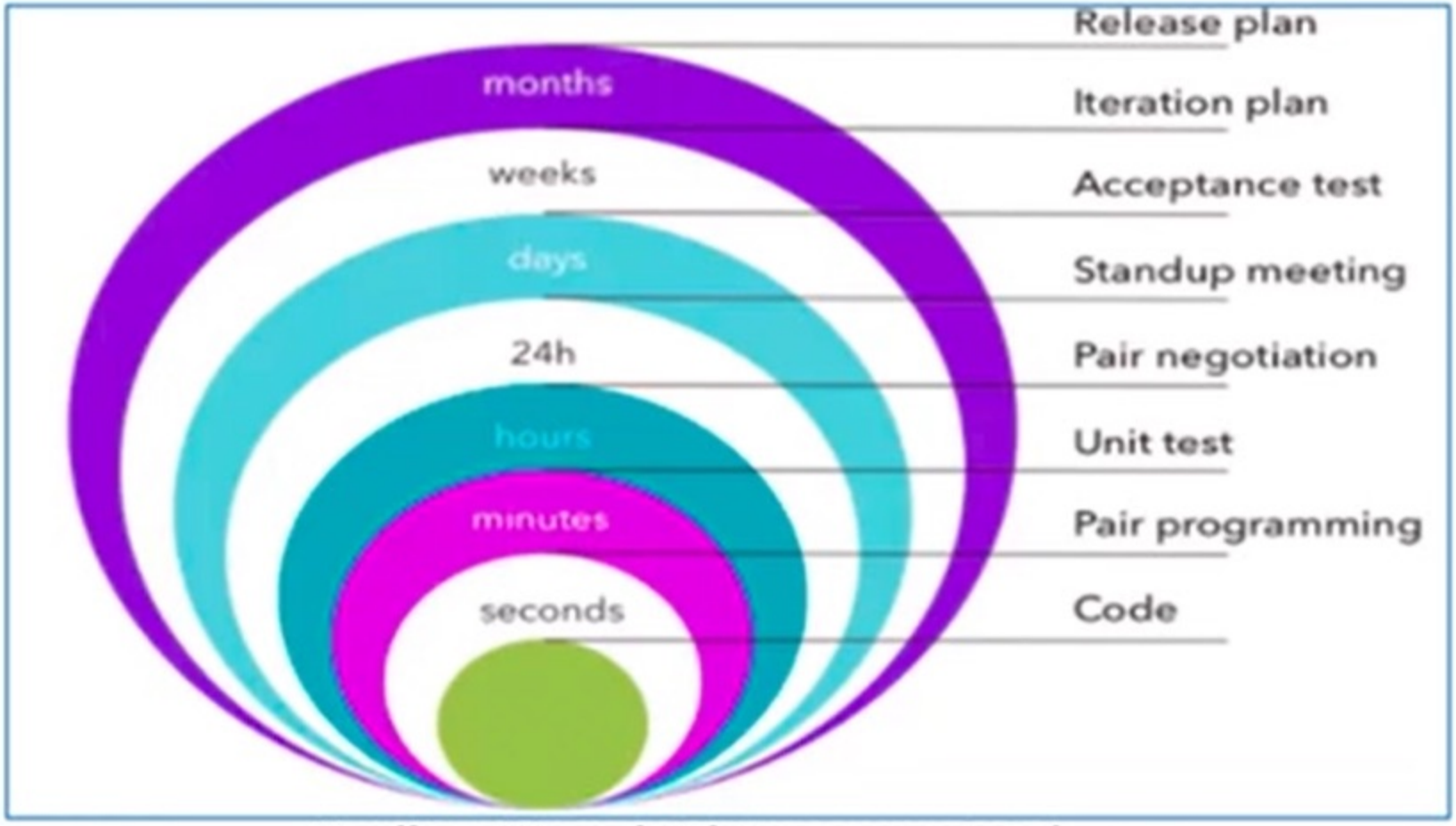
- Coach
  - This  role is helpful if the team is just starting up
  - Coach plays a supporting role in the team's success
  - Coach is usually an outside assistant or an external consultant
  - Watches the team's work and teaches the members to implement the most effective practices
  - Ensures that the project stays on track
  - Sets up a work culture that's balanced with the rest of the organization
- Responsibilities of Coach
  - Identifying the Extreme programming practices that help resolve the problems
  - Always being ready and  helping the team members
  - Overseeing the work

# Roles And Responsibilities (Continued …)

TEKNOTURF

- Tracker
  - This role is optional
  - This role can be created on a need-only basis
  - This is carried out by one of the developers to track of team's progress metrics
  - Some of these metrics may include the amount of time worked, amount of overtime, passing and failing tests and velocity
  - Links customers and developers
  - Communicates with every team member to identify roadblocks and to figure out solutions
  - Determines how many stories are to be completed and ensures that the stories are completed within a given iteration
- Responsibilities of Tracker
  - Organizing meet-ups
  - Regulating discussions
  - Keeping track of important progress KPIs

# Feedback Loops




https://www.digite.com/agile/extreme-programming-xp/

# Feedback Loops (Continued ...)

TEKNOTURF

- Unit Test
  - It is the first Feedback loop that should be established on any project
  - Testing gives feedback on your work in seconds
  - You can have this feedback every time you save a code file or when you push the code in the version control system
- Pair Programming
  - Pair programming is a powerful loop technique that provides peer feedback in seconds
  - It lets you consolidate your knowledge and explain problems or solutions to someone else
  - Developers receive feedback constantly by working in pairs and testing code as it is written

# Feedback Loops (Continued …)

TEKNOTURF

- Code Review
  - Code review definitely improves project quality
  - It should be performed every time a developer in the team makes a pull request directly inside the version control system
  - It takes days, not seconds or minutes to provide the feedback
- Daily Stand-up
  - This ceremony provides an opportunity to synchronize team member efforts as well as a starting point for discussing potential improvements
  - It is a daily activity whose value hinges not only on the technical aspects but also in team communication and the big picture behind the project
  - Managers get feedback on progress and obstacles at the daily stand up meeting
- Acceptance Test
  - Customers get feedback on progress with acceptance test scores and demonstrations proceeding every iteration

# Continuous Process

TEKNOTURF

- Continuous Integration
  - It is an Extreme Programming practice where members of a delivery team frequently integrate their work hourly, or at least once daily
  - Each integration is verified by an automated build, which also performs testing, to detect any integration errors quickly and automatically
- Refactoring
  - Refactoring is the technique of improving code without changing functionality
  - It is an ongoing process of simplification that applies to code, design, testing, and XP itself
  - A repeatable process is needed to guide refactoring
  - Refactoring is especially necessary for Extreme Programming because they require strong customer involvement in the working process
- Small Releases
  - XP promotes small Releases through continuous integration and other extreme programming practices
  - Small Releases helps to deliver a small working increment (User Story) in a weeks time
  - Development Team can do a weekly review

# Extreme Programming Versus Other Frameworks

TEKNOTURF

- Extreme Programming Versus Scrum
  - Scrum has sprints that are 2 to 4 weeks long, while XP iterations are shorter taking 1 to 2 weeks
  - Extreme Programming is much more flexible with possible changes within iterations, while Scrum doesn't allow any modifications after the sprint backlog is set
  - In XP, the customer prioritizes features and decides on the order of their development, but in Scrum, the team itself determines what to work on first
- Extreme Programming Versus Kanban
  - Kanban puts a lot of focus on visualizing the development process and strictly limits the number of features developed at a time
  - Kanban is also characterized by a continuous workflow while XP has separate iterations

# Extreme Programming Versus Other Framework (Continued …)


- Extreme Programming versus Dynamic System Development Method(DSDM)
  - In Extreme Programming, the Customer writes/collects stories, whereas in DSDM, facilitated workshops identify high level requirements
  - In XP, a group of developers estimate the size of a story in Days/Weeks/Months, while in DSDM, the Project Manager and Developers estimate effort derived from size using function points

# Agile Methodology

- Agile methodology is an approach to project management that uses four values and twelve principles to organize projects
- It works in ongoing sprints of project planning and execution
- Agile projects require an iterative approach
- It has a high level of customer involvement
- You can run an Agile project using several different frameworks:
  - Scrum
  - Kanban
  - Extreme Programming (XP)
  - Dynamic System Development Method (DSDM)
  - Feature Driven Development
  - Lean Software Development