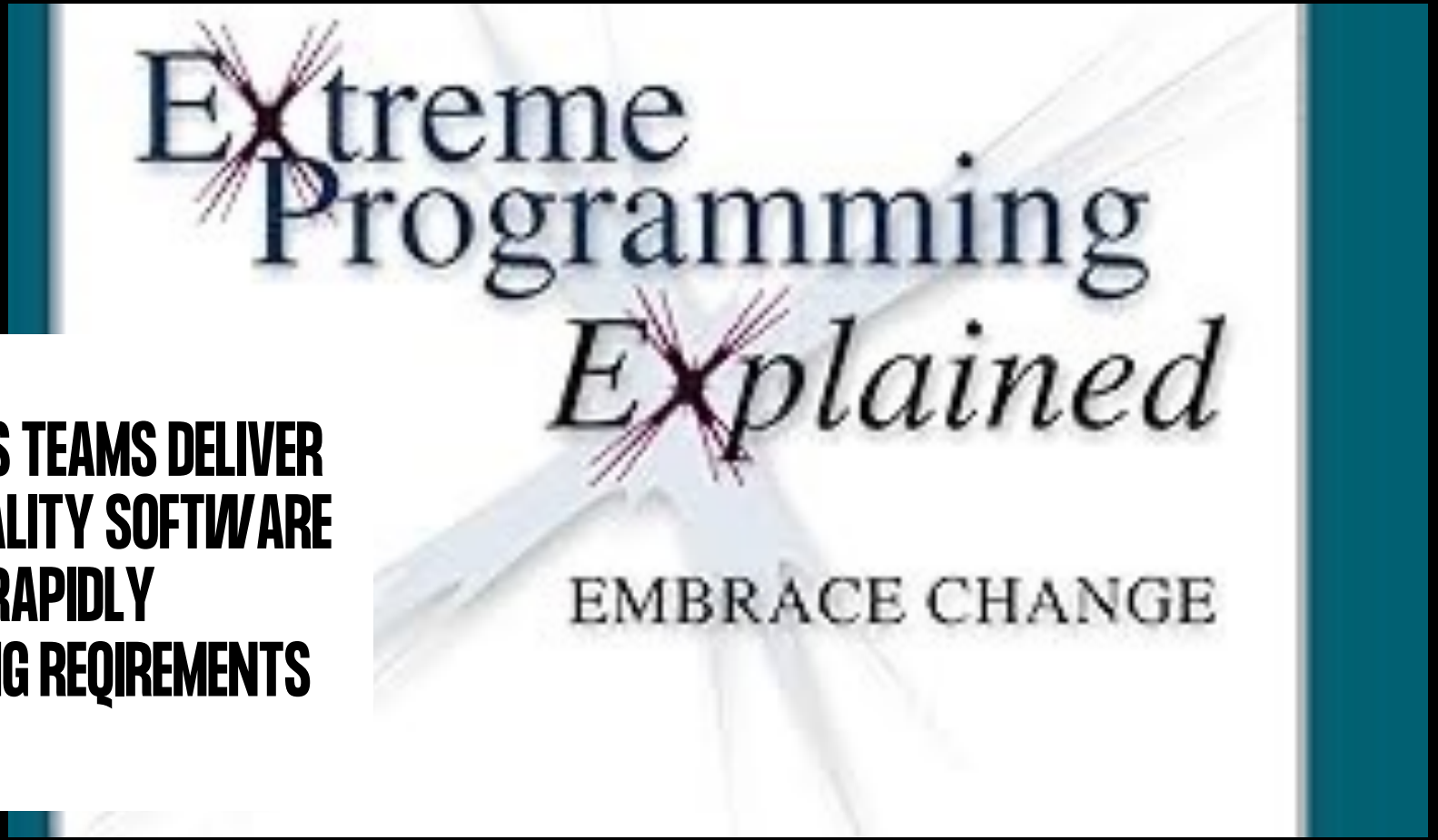


XP HELPS TEAMS DELIVER HIGH QUALITY SOFTWARE ADMIST RAPIDLY CHANGING REQIREMENTS

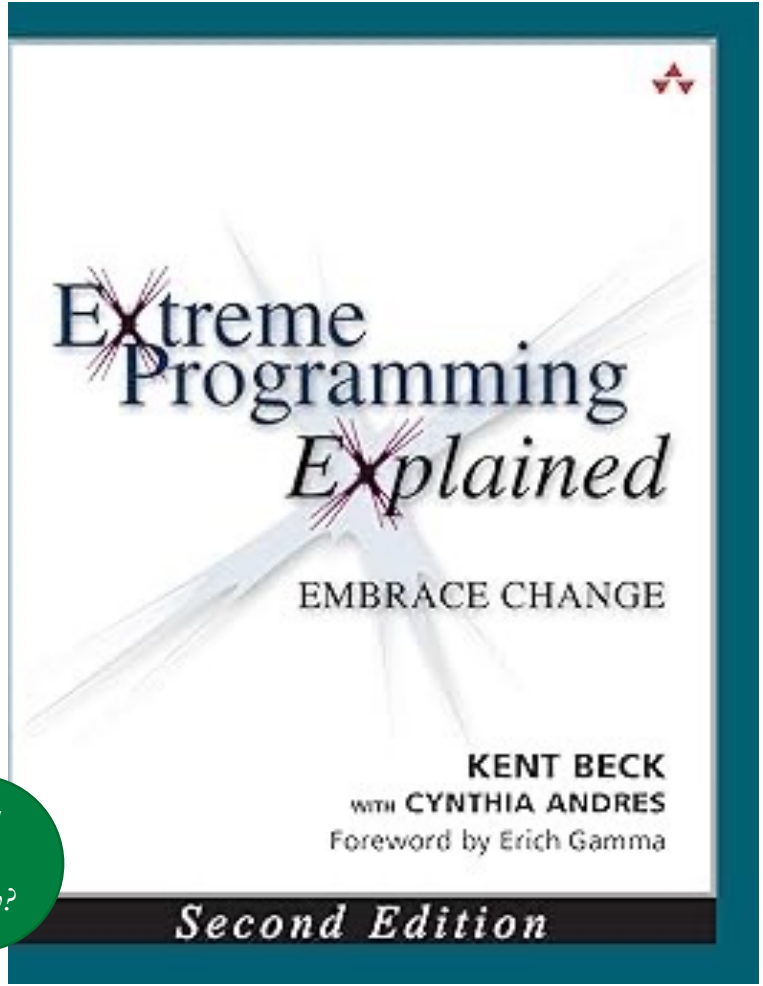
BRIEF HISTORY

This deck is based on Kent Beck & Cynthia Andres book Extreme Programming (XP) Explained: Embrace Change.

Kent Beck pioneered practices including test driven development and was one of the original signatories of the Agile Software Development Manifesto in 2001.

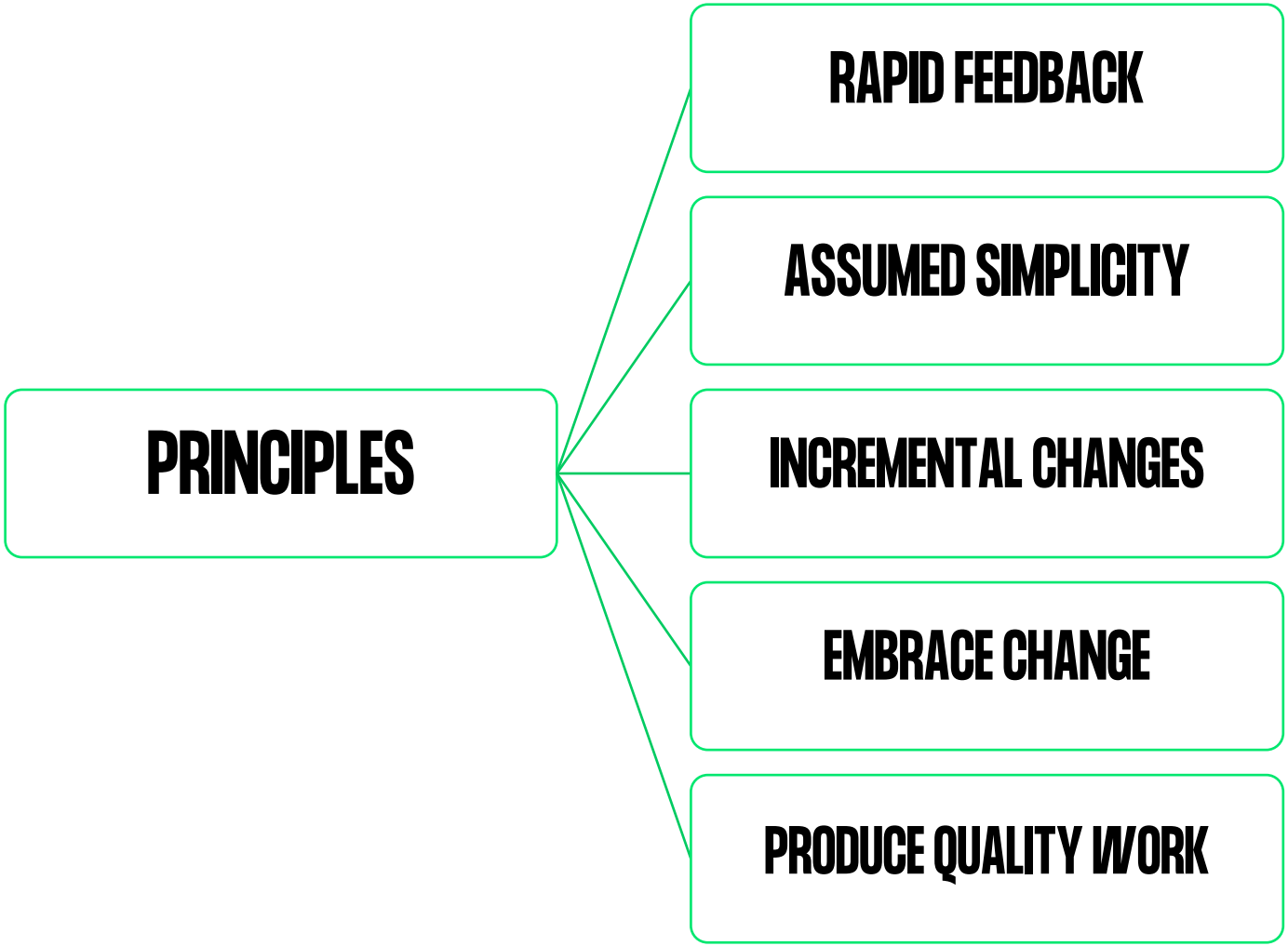
XP turned 25 this year!

Did you know?



XP HAS 5 VALUES

- Communication
- Feedback
- Simplicity
- Courage
- Respect

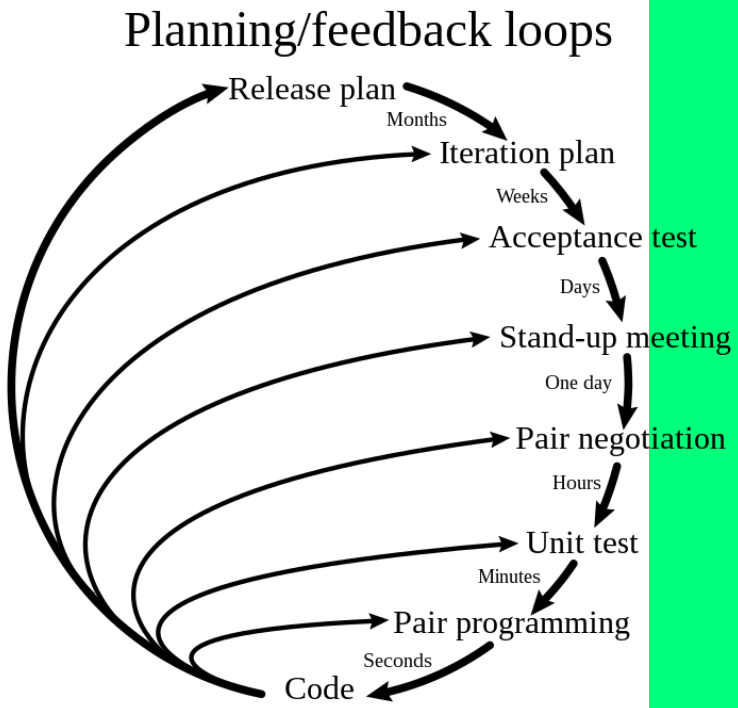

PRINCIPLES
RAPID FEEDBACK
ASSUMED SIMPLICITY
INCREMENTAL CHANGES
EMBRACE CHANGE
PRODUCE QUALITY WORK

RAPID FEEDBACK

XP emphasises the importance of getting feedback as quickly as possible and incorporating it into the development process. This principle aligns with:

- Feedback (directly tied to one of XP's core values).
- Flow (ensuring a steady sustainable pace that allows for frequent feedback loops).
- Reflection (taking the time to analyse and act on feedback to improve).
- Improvement (continuous feedback helps identify areas for enhancement).
- Opportunity (viewing feedback as a chance to refine the system).
- Failure (accepting that failure is a source of valuable feedback).

These principles connect with the idea that frequent feedback helps guide development and ensures that the team can make real-time adjustments to deliver better software.



ASSUME SIMPLICITY

XP stresses simplicity in both design and decision-making, ensuring that you always choose the simplest solution that works. This principle incorporates:

- **Simplicity** (one of XP's core values, advocating for straightforward, simple designs).
- **Economics** (by focusing on simplicity, development is more cost-effective).
- **Self-Similarity** (simple solutions that scale across different levels of complexity).
- **Redundancy** (building safeguards into the process to ensure simplicity and robustness).

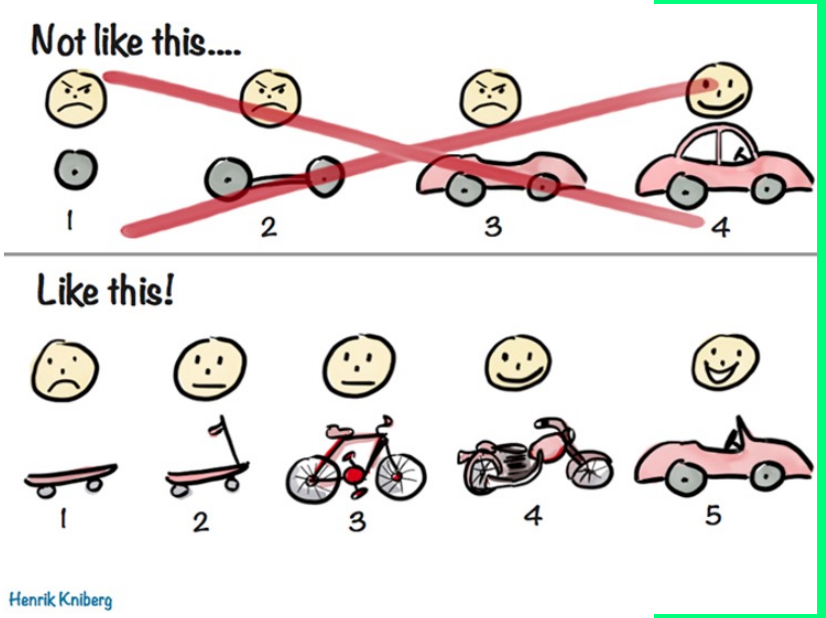
These elements tie into the belief that **simple solutions are often the most effective**, and overcomplicating things increases risk and technical debt.

INCREMENTAL CHANGES

XP supports making small, manageable changes that allow the team to adapt and evolve the software over time. This aligns with:

- **Baby Steps** (taking small, incremental steps to minimize risk and maximize flexibility).
- **Flow** (maintaining a steady pace by delivering incremental improvements).
- **Economics** (delivering in increments allows the business to see ROI earlier).
- **Feedback** (incremental delivery provides regular opportunities for feedback and adjustment).

These principles reflect the approach of **gradual, continuous improvement**, making it easier to adapt to new information or requirements without large-scale disruptions.



EMBRACE CHANGE

Change is inevitable, and XP encourages teams to be courageous and proactive in responding to it. This principle includes:

- **Courage** (one of XP's core values, emphasizing the importance of embracing risk and uncertainty).
- **Opportunity** (viewing change as a chance to improve rather than a threat).
- **Failure** (accepting that failure is part of the process of adapting to change).
- **Self-Similarity** (ensuring solutions are flexible enough to handle change at multiple scales).

These principles underscore the idea that **change is a constant**, and XP teams should be prepared to embrace it confidently, making decisions that support future flexibility.

HIGH QUALITY WORK

High-quality work is fundamental to XP, ensuring that all stakeholders benefit from the process and that the software remains maintainable and scalable. This principle encompasses:

- **Quality** (directly tied to delivering software that works and can be easily maintained).
- **Humanity** (recognizing that quality comes from a humane, supportive environment).
- **Mutual Benefit** (creating value for all parties involved — customers, developers, and businesses).
- **Accepted Responsibility** (team members taking ownership of the quality of their work).
- **Diversity** (including different perspectives ensures comprehensive solutions).

The emphasis on **high-quality work** ensures that everyone involved in the project, from developers to customers, gains value from the process, and the software remains robust and maintainable.

AND 12 PRIMARY PRACTICES + COROLLARY ONES

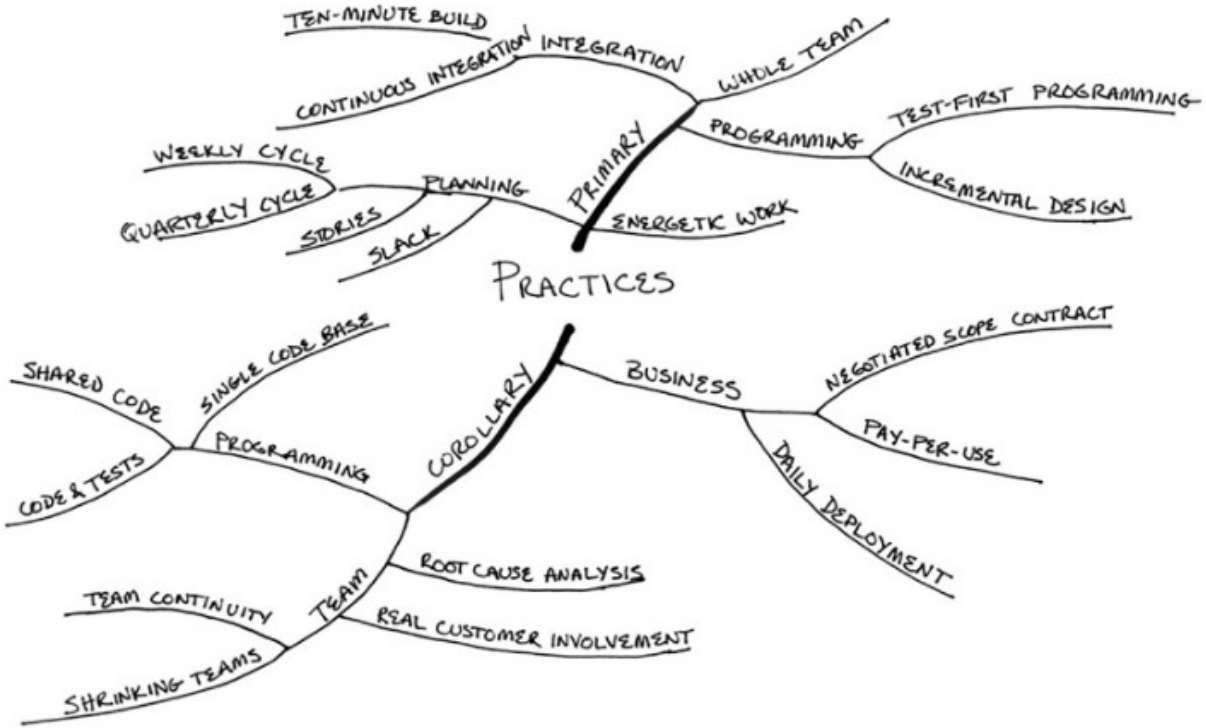
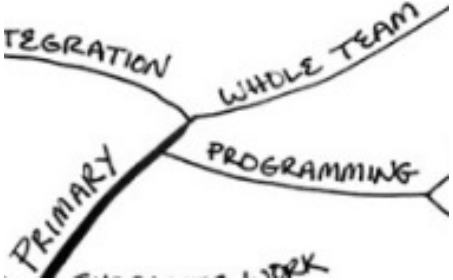


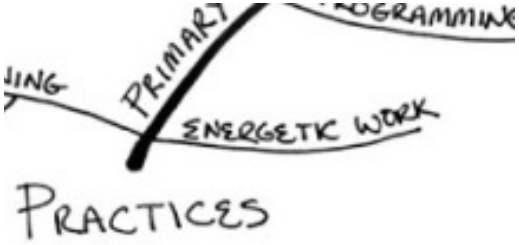
FIGURE 3. Summary of practices

WHOLE TEAM

- **Sit Together:** Develop in an open space, big enough for the whole team.
- **Whole Team:** Include on the team people with all the skills and perspectives necessary for the project to succeed.
- **Informative Workspace:** Make your workspace about your work. Visualise WIP.



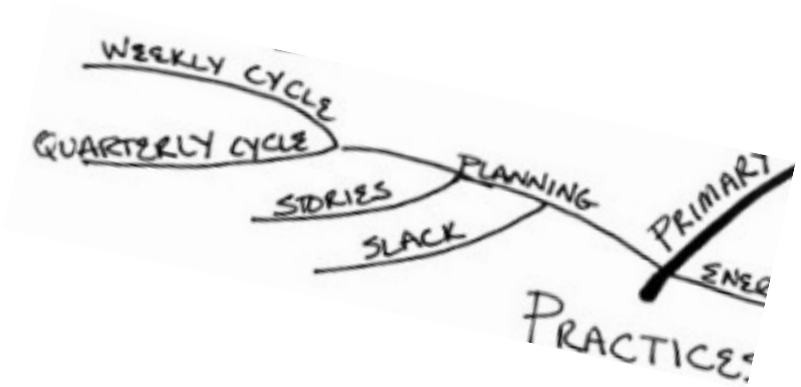
ENERGETIC WORK



- **Energized Work:** Work only as many hours as you can be productive and only as many hours as you can sustain. Refers to maintaining a sustainable pace, ensuring the team stays productive without burnout.
- **Pair Programming:** Write all production programs with two people sitting at one machine.

PLANNING

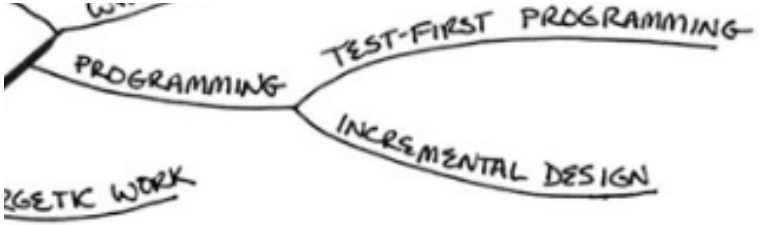
- **Stories:** Represents user stories that guide the planning process.
- **Weekly Cycle:** Involves planning work in weekly iterations.
- **Quarterly Cycle:** Involves planning for longer-term objectives, typically over a few months.
- **Slack:** Involves including some flexibility in the planning to handle unexpected issues.



INTEGRATION

- **Ten-Minute Build:** Ensures that the build process is quick, ideally taking no more than 10 minutes.
- **Continuous Integration:** Involves frequently integrating and testing the codebase to detect problems early.

PROGRAMMING



- **Test-First Programming:** Represents Test-Driven Development (TDD), where tests are written before the actual code.
- **Incremental Design:** The system design evolves gradually as new features are added.
- **Single Codebase:** The codebase is owned collectively by the whole team, meaning anyone can improve any part of the code at any time. This practice promotes responsibility and reduces bottlenecks.

XP is giving up old, ineffective technical and social habits in favor of new ones that work

XP is giving up old, ineffective technical and social habits in favor of new ones that work

XP is fully appreciating yourself for your total effort today

XP is giving up old, ineffective technical and social habits in favor of new ones that work

XP is fully appreciating yourself for your total effort today

XP is striving to do better tomorrow

XP is giving up old, ineffective technical and social habits in favor of new ones that work

XP is fully appreciating yourself for your total effort today

XP is striving to do better tomorrow

XP is evaluating yourself by your contribution to the team's shared goals

XP is giving up old, ineffective technical and social habits in favor of new ones that work

XP is fully appreciating yourself for your total effort today

XP is striving to do better tomorrow

XP is evaluating yourself by your contribution to the team's shared goals

XP is asking to get some of your human needs met through software development