References for report

Definition of extreme programming:

It is a software development method that views people, rather than paper, as a project’s most potent element. (Martin, 2000)

Changeability :

In XP, developers initially document requirements through user stories, which are basically textual use-case descriptions (see the “Common XP Terms” sidebar). To clarify these requirements and set priorities, XP uses an on-site customer representative who works with the team. This practice improves the software’s business value: When issues arise, programmers can get customer input immediately rather than speculate on customer preferences. (F. Maurer, 2002)

A short release cycle also helps developers deal with changing requirements and reduces the impact of planning errors. (F. Maurer, 2002)

before adding code to the system, the programmers must write a failing unit test that the new code must make successful. (Martin, 2000)

Complexity:

The customer defines functional (acceptance) tests, which the development team implements. From a business perspective, these tests verify that the program does what it is supposed to do. (F. Maurer, 2002)

XP uses patterns, but it does so according to a simple idea: Rather than implement with patterns, developers should “refactor to patterns when appropriate and away from patterns when something simpler is discovered.”8 Constant refactoring ensures that the design is always as simple as possible. (F. Maurer, 2002)

Two heads are better than one for achieving correctness on highly complex programming tasks. They might also have a time gain on simpler tasks. (Tore Dybå, 2007)

Invisibility:

It’s much easier to accurately estimate a task when you know you won’t be spending much time debugging. It’s much easier to know and communicate a task’s status when you can see a fraction of the tests passing. (Martin R. C., 2007)

Conformity:

Be communicative with everyone on the team—including customers, end users, and business folks. Consistently and doggedly strive to understand and deliver what your customers want with the highest possible quality (Williams, 2003)

XP programmers gather requirements as short, natural-language statements that are essentially the customer’s words written on small index cards. Called user stories, these cards are commitments for further conversation between the customer and developers and are not intended to completely specify requirements. (Williams, 2003)

Links:

<https://ieeexplore.ieee.org/document/854062>

<https://ieeexplore.ieee.org/document/989006/authors#authors>

<https://ieeexplore.ieee.org/document/4375233/authors#authors>

<https://ieeexplore.ieee.org/document/1196315>

<https://ieeexplore.ieee.org/document/4163026>