Instituto Tecnológico de Tijuana

Ingeniería en Sistemas Computacionales

Investigación I:

Pair Coding

Materia: Mineria de Datos

Unidad: Unidad I

Facilitador:

José Christian Romero Sánchez

Alumno: Hernández Negrete Juan Carlos

Fecha:

Tijuana Baja California a 16 de Marzo del 2021

Pair Coding

Definition

Pair programming consists of two programmers sharing a single workstation (a screen, keyboard, and mouse between the pair). The programmer at the keyboard is usually called the "controller", the other, also actively involved in the programming task, but more focused on the general direction is the "browser"; Programmers are expected to switch roles every few minutes or so.

Common mistakes

- Both programmers must actively participate in the task during a paired session; otherwise, no benefit can be expected.
- A simplistic but often raised objection is that matching “doubles the costs”; That's a misconception based on equating programming with typing; however, one should keep in mind that this is the worst result of a misapplied pairing
- At least the driver, and possibly both programmers, are expected to keep an active comment; Pair programming is also "programming out loud": if the driver is silent, the navigator must intervene
- Partner programming cannot be fruitfully imposed on people, especially if relationship problems, including more mundane ones (such as personal hygiene), get in the way; Solve these first!

Origins

The names of various celebrities have been invoked in an attempt to imbue pair programming with an aura of necessity if not sanctity; anecdotes of John Von Neummann, Fred Brooks, Jerry Weinberg, Richard Gabriel or Edsger Dijkstra using the practice are fascinating but sometimes hard to substantiate. However, the following timeline of verifiable sources does suggest that pair programming, in its modern form, has been around since well before the Agile movement:

- 1992: “Dynamic Duo” is the term coined by Larry Constantine, reporting on a visit to Whitesmiths Inc., a compiler vendor started by P.J. Plauger, one of the implementors of C: “At each terminal were two programmers! Of course, only one programmer was actually cutting code at each keyboard, but the others were peering over their shoulders.” Whitesmiths existed from 1978 to 1988.
- 1993: “The benefits of collaboration for student programmers” by Wilson et al. is one early empirical study indicating benefits of pairing for programming tasks specifically. Posterior studies are more abundant and driven by the desire to “validate” pair programming after it had already gained popularity through Extreme Programming.
- 1995: the pattern “Developing in Pairs” is given a brief description, in Alexandrian pattern form, in Jim Coplien’s chapter “A Generative Development-Process Pattern Language” from the first patterns book, “Pattern Languages of Program Design”.
- 1998: in “Chrysler goes to Extremes”, the earliest article about Extreme Programming, pair programming is presented as one of the core practices of the C3 team; it is later described formally as one of XP’s original “twelve practices”

- 2000: (or earlier) – the roles of Driver and Navigator are introduced to help explain pair programming; the earliest known reference is a mailing list posting; note however that the reality of these roles has been disputed, for instance Sallyann Bryant's article "Pair programming and the mysterious role of the navigator"
- 2002: "Pair Programming Illuminated", by Laurie Williams and Robert Kessler, is the first book devoted exclusively to the practice and discusses its theory, practice and the various studies up to that date
- 2003: an anonymous article on the C2 Wiki describes Ping-Pong Programming, a moderately popular variant which marries pairing with test-driven development.
- 2015: James Coplien publishes Two Heads are Better Than One which provides an overview of the history of Pair Programming that traces its origins back to the mid 1980's if not before.

Skill levels

As suggested above, one of the main problems preventing effective matching is passivity. When used concurrently with test-based development, a variant called "ping-pong programming" encourages more frequent role switching: One programmer writes a unit test that fails, then passes the keyboard to another who writes the corresponding code, then goes to a new test. This variant can be used for educational purposes only, or by experienced programmers as a playful variant.

Beginner:

o Able to participate as a navigator, in particular to intervene appropriately

o Able to participate as a driver, in particular to explain the code while writing it o Intermediate

o Can say the right time to leave the keyboard and switch roles

o You can tell the right time to "steal" the keyboard and switch roles

Advanced:

- *Able to "drop in" when another pair has been working on a task and pick up the navigator role smoothly*

Expected benefits

- Higher code quality: "Program out loud" leads to clearer articulation of the intricacies and hidden details in coding tasks, reducing the risk of errors or falling into dead ends.
- Better dissemination of knowledge among the team, particularly when a developer who is not familiar with a component is paired with one who knows it much better.
- Better transfer of skills, as junior developers acquire micro-techniques or broader skills from more experienced team members.
- Great reduction in coordination efforts, as there are N / 2 pairs to coordinate instead of N individual developers
- Improved resistance of one pair to interruptions, compared to an individual developer: when one member of the pair must attend to an external prompt, the other can remain focused on the task and can help to regain focus afterwar

References

- Desconocido. (2019). Pair Programming: Does It Really Work?. 2020, de Agile Alliance Sitio web: https://www.agilealliance.org/glossary/pairing/#q=~(infinite~false~filters~(postType~(~'page~'post~'aa_book~'aa_event_session~'aa_experience_report~'aa_glossary~'aa_research_paper~'aa_video)~tags~(~'pair*20programming))~searchTerm~'~sort~false~sortDirection~'asc~page~1)

- Javier Garzas. (25 de Junio del 2012). ¿Beneficios del pair programming? ¿Dos programadores en un solo ordenador es perder medio equipo?. 2020, de 233Academy.com Sitio web: https://www.javiergarzas.com/2012/06/beneficios-pair-programming.html