



EDUCACIÓN
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO
NACIONAL DE MÉXICO®



**Tecnológico Nacional de México
Instituto Tecnológico de Tijuana**

**Subdirección Académica
Departamento de Sistemas y Computación**

Semestre:

Febrero – Junio 2021

Carrera:

Ingeniería en Tecnologías de la Información y Comunicaciones

Materia y serie:

Minería de datos

BDD-1703TI9A

Unidad a evaluar: Unidad I

Nombre de la Tarea:

Pair coding

Nombre del Alumno:

Manuel Javier Sifuentes Martinez 17212934

Nombre del docente:

José Christian Romero Hernández

Pair Coding

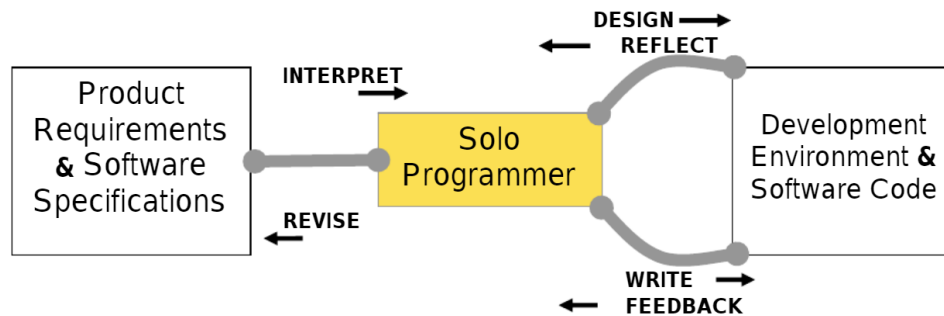
Pair programming, for the purposes of this study, describes a programming technique where all programming work is done by two programmers, working together at a single PC. Within the pair, work is split into two roles, known as the driver and the navigator. The driver is the person at the keyboard, responsible for the actual typing of the code being generated. The navigator is an active observer and monitor of the code being written. The driver and navigator collaborate on all aspects of the software development: design, coding, debugging, etc. They are in constant communication, asking and answering questions of each other. The two programmers may switch roles frequently in the course of a programming session.

The simplest view of why pair programming works is that two people make better design decisions than one. This view characterizes programming as a series of design decisions that are translated into code. The presence of a second individual distributes the cognitive task of programming, aiding design discussion and error finding. More specifically, working in pairs has the following influences on decision-making:

- Two individuals will have overlapping, but not identical, sets of information. When working together as a pair, sharing this increased pool of information can lead to better decision-making.
- Design collaboration affords a mutual apprenticeship, where through the collaboration each participant learns some of the technical skills and methods of their collaborator.
- Collaborative design requires the negotiation of a shared understanding and mutual orientation. This negotiation process makes explicit the cognitive processes that are normally tacit when working individually.
- This negotiation process requires that programmers produce an account of goals, plans, decisions and actions. This appears to lead to a more thorough exploration of design options. This account production, verification, and affirmation leads to increased confidence by the programmers and vets flawed design ideas earlier.

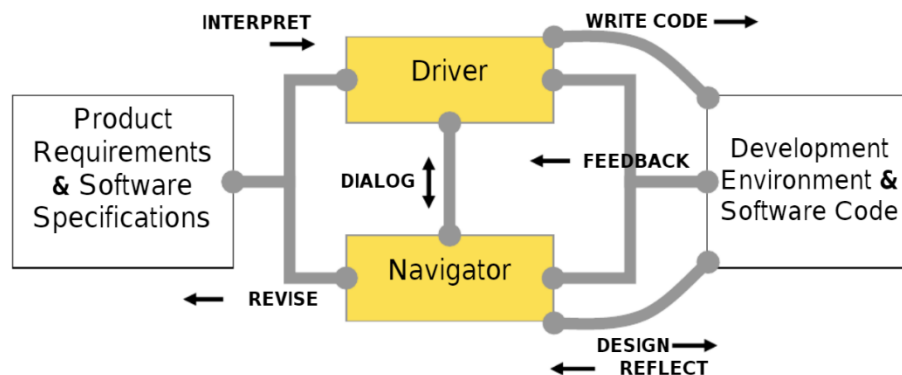
Working in pairs also has influences when design decisions are translated into code. By monitoring the coding, the navigator can look for missed cases and typographical errors. The navigator can also think ahead of the code being typed at a given moment. One way of stating this is that the navigator can consider issues that have a longer time constant than those being addressed by the driver.

Solo programming is characterized as follows:



The solo programmer, in addition to interpreting and revising the specifications, must attend to the code at many levels, ranging from high-level design and design revisions to low-level entering of program statements and understanding of debugging results.

For pair programmers, the situation is different:



In the pair programming software development paradigm, the driver and navigator act on the specifications in tandem and develop code. The two actors alternate roles frequently during the task. In this alternating dynamic, we postulate that the navigator's focus on higher level conceptual relationships and goals allow him/her to take on a coaching role where he/she observes the driver's interaction with the code, identifies needs and opportunities, and intervenes to supply needed information and/or strategy to arrive at the desired goal, while the driver attends to the immediate coding task at hand. Inclusion of the navigator (coach) introduces multiple feedback paths for knowledge creation and error correction. This view is consistent with our earlier statement that the navigator is performing tasks with a longer time constant than those of the driver.

Pair Programming Advantages

- **More Discipline.** By pairing correctly they are more likely to do "the right thing" rather than taking long breaks.
- **Better code.** Matching similar is less likely to produce bad designs since your intention tends to design with higher quality.
- **Constant work flow.** Pairing produces a different workflow when working alone. In a Relationship, the workflow recovers more quickly: one programmer asks the other "where do we meet?" Pairs are more resistant to interruptions as one developer deals with the interruption while the other continues to work.
- **Multiple new developments to the design.** If the pairs rotate frequently in the project it means that more people are involved with a particular characteristic. This helps create better solutions, especially when a couple cannot solve a difficult problem.
- **Improved morale.** Pair programming is more enjoyable for some programmers than programming alone.
- **Collective ownership of the code.** When done in pairs, and the pairs rotate frequently, everyone has a knowledge of the codebase.
- **Teaching.** All have knowledge that the others do not. Pair programming is a fun way to share knowledge.
- **Team cohesion.** People get to know each other more quickly when programming as a couple. Pair programming can enliven the team feeling.
- **Few interruptions.** People are more reluctant to interrupt a partner than a person who works alone.
- **Fewer workstations.** Since two people will be working at one workstation, fewer workstations are required, and the extra workstations can be occupied for other purposes.
- **Two heads are better than one.** If the driver finds a problem with the code, there will be two of them that will solve the problem.
- **Fewer coding errors.** Since there is another programmer reviewing his work, the result is better code. In addition, it allows the driver to stay focused on the code being written while the other handles external issues or interruptions.
- **Develop the interpersonal skills of your staff.** Collaborating on a single project helps your team appreciate the value of communication and teamwork.

References

Chong, J., Plummer, R., Leifer, L., Klemmer, S., Eris, O., & Teye, G. (2005) Pair programming: When and why it works. March 16, 2021. Web site: https://www.researchgate.net/publication/228653369_Pair_programming_When_and_why_it_works

Stackify. (s.f) What is Pair Programming? Advantages, Challenges, Tutorials & More. March 16, 2021. Web site: <https://stackify.com/pair-programming-advantages/>