

# “Snapshot” of the State of Software Reuse in Colombia

Luisa Restrepo Gutierrez<sup>a</sup>, Raúl Mazo Peña<sup>a,b</sup>, Daniel Correa Botero<sup>a</sup>, Paola Vallejo Correa<sup>a</sup>,  
Elizabeth Suescún Monsalve<sup>a</sup>

<sup>a</sup>Universidad EAFIT, Medellín, Colombia

<sup>b</sup>ENSTA Bretagne, Brest, France

---

## 1. SURVEY RESULTS

This section presents the survey results and the answer to the 26 research questions.

Fig. 1 shows the number of participants of the survey grouped by Colombia regions. The software organizations that answered the survey were mainly located in the Colombian Middle-East region because 64.4% of software companies are located in this region. The Coffee-belt region follows this with 18.2% of the survey population, the Pacific region with 9.5%, and the other regions with 7.9% of the survey population. Regarding the answer per region, 12% was the average coverage, where Middle-East was a 5%, Pacific region with 6%, South-center with 8.1%, Coffee-belt with 9.4%, the Caribbean with 9.7%, and Llanos with 33.3% of coverage.

Llanos with 40%, Coffee-belt with 67%, Middle-east with 72%, Pacific with 73%, Caribbean with 76%, and South-center with 80% reported a reuse success level that ranges from strong to substantial, which indicates that Colombian software organizations are at similar levels in terms of software reuse (68% on average) regardless of the region where they are located. It is stressed that 68% of the participants indicated that reuse success is associated with companies that measure the level of reuse; the remaining companies made subjective analysis.

The participants represented a broad range of software domains (see Table 1). The categories are not mutually exclusive, and one participant can be placed in multiple software domains. Most of the participants work in software companies dedicated to developing custom products.

Table 2 presents the participant roles. The categories are not mutually exclusive, and one participant can play multiple roles. Most of the participants had the CIO role as their main role while playing a complementary role. In Colombia, mainly in Startups, the CIO can hold multiple roles, as back-end, front-end software developer, and architect at the same time. In large, medium, and small companies, the organizational structure is very well-defined, so people, in general, do not occupy more than one role.

### 1.1. RQ1- What are the characteristics of software production (business factors)?

This section presents an analysis of how the business factors influence software reuse success. This analysis is made based on the survey questions' responses, related to research questions, related to the business factors.

#### 1.1.1. Application domain

Fig. 2 shows the application domains in which software development companies in Colombia work and the associated reuse success level, which helps identify application domains where

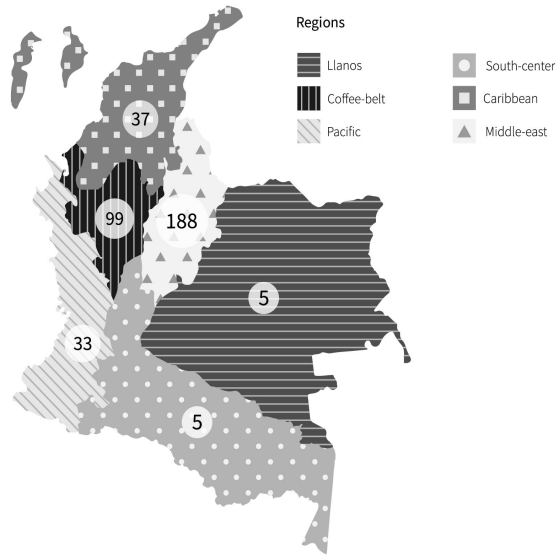


Figure 1: Software organizations participants by regions of Colombia

software reuse is more applied. According to the data collected, the following results were obtained:

- Most of the organizations work for the Information and Communications Technologies (ICT) domain (49% of the respondents). In the category Other, the participants indicated to work in Transport, Defense, and Aviation domains.
- The 3% of the organizations reported working for all application domains because the company focuses on developing custom software regardless of the domain.
- Telecommunications domain reported the highest reuse success level (46% for Substantial level and 32% for Strong level), coinciding with Frakes (1) who conducted a reuse study in which Telecommunications domain presented significant levels of reuse.
- For all application domains, most answers about the level of reuse success were Substantial and Strong. Legal domain was the only application domain without reporting No success in software reuse. The health and social security domain reported the highest No success level (15%), attributed to legal issues.

The results indicate that are application domains that affect more software reuse success. The regression test result confirms it because it concludes that communications, construction and engineering, financial, legal, logistics, manufacture, health, and social security, insurance, telecommunications, tourism, and entertainment domains have the most influence on the software reuse success, answering research question **RQ01**.

#### 1.1.2. Type of software developed

To identify the influence of the type of software developed in the reuse success, survey participants answered a multiple-choice question about the type of software built in their companies.

Table 1: Population's software domains

Software domain	Percentage
Custom software development	68.1%
Development of new products	48.2%
Software and application integration's	48.2%
Computer consultancy activities	47.4%
Software maintenance	36.2%
In house development	35.1%
Administration of computer facilities	24.8%
Software testing	24.5%
Hosting	21.3%
IT (Information Technology) training programs	20.2%
SaaS (Software as a Service)	2.4%

Table 2: Participant roles

Role	Percentage
Chief Information Officer (CIO)	31.3%
Software development analyst	24.8%
Back-end software developer	24.3%
Software architect	21.0%
Front-end software developer	19.3%
Chief Technology Officer (CTO)	14.2%
Database administrator	13.9%
Consultant	10.9%
Component's developer	9.5%
Test analyst	8.4%
Business Analyst (BA)	6.3%
Requirements engineer	6.0%
Other	5.7%
Coach	1.9%

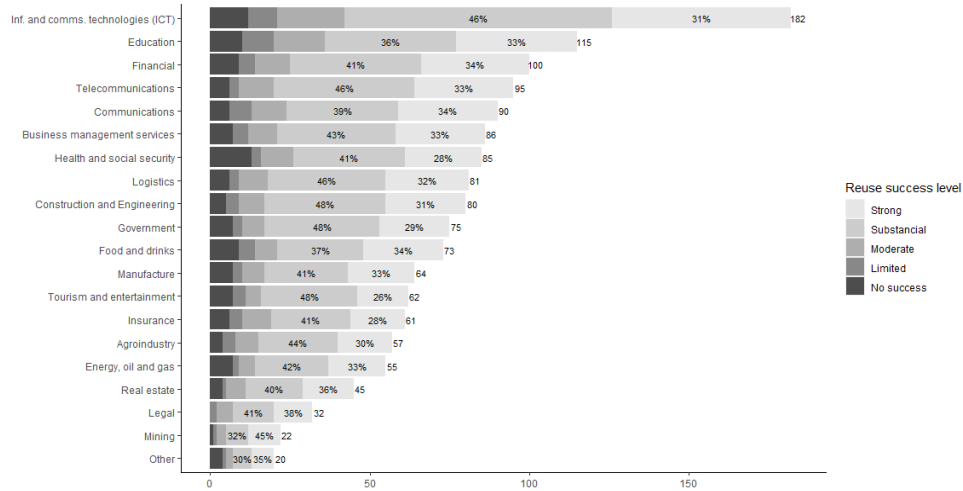


Figure 2: Applications domains and reuse success

Answers are not mutually exclusive, and one participant can select multiple types of software developed.. According to the data collected, the following results were obtained:

- The most developed type of software developed (Web applications), it is complemented with the development of other types of software.
- Participants from Micro-enterprise and Small organizations indicated to develop Integration software (in category Other). Most Startups and Small companies work on Virtual and/or augmented reality software.
- All organizations, regardless of their size, develop any type of software.
- Strong and Substantial reuse success level for all types of software was between 70% and 78% for most companies; this indicates that reuse success applies to all types of software. Software integrated into electrical devices (81%) had greater success. Video games (63%) reported the lowest reuse success level but still have a strong and substantial reuse success.

The results indicate that there are types of software developed that affect more software reuse success. The regression test result confirms it because it concludes that applications for mobile devices, web applications, web applications for data analytics, digital content, information systems for business management, IoT systems, solutions based on artificial intelligence, and virtual and/or augmented reality have the most influence on the software reuse success, answering research question **RQ02**.

#### *1.1.3. Product family approach*

A product family, also known as a software product line (SPL), can be defined as "a set of software-intensive systems sharing a common, managed set of features that satisfy the specific needs of a particular market segment or mission and that are developed from a common set of core assets in a prescribed way" (2). Software product line engineering (SPLE) is a growing area showing promising results in research and practice. It is further development and acceptance in industry (3). For taking advantage of this, it is important to know the concept, 42% of the companies were familiar with this term.

According to the data collected, the following results were obtained:

- The 49% of the companies were dedicated to developing a product, this product was the priority, and the main profit provider of the company, 13% of these companies also developed a family of products, due to the expertise, they also developed similar products for other companies as a second income or the same company.
- The 48% of the companies develop more than one software product with common characteristics, but only 14% focus on a single application domain.
- The 29% of the companies develop products that have little or nothing in common, these companies reported the lowest strong and substantial reuse success with a total of 56%, against 67% of strong and substantial reuse success reported for companies that develop one product and 77% for companies that develop a family of products.

According to these results and regression test results, using a particular reuse approach as a product family approach influences the software reuse success, answering research question **RQ03**.

#### 1.1.4. Domain analysis

Before creating a product, the market analysis should be made to identify the domain to be modeled (or what COTS to be used) to develop reusable software components to be placed in the software reuse repository (4; 5; 6), this experience provides knowledge of commonalities and variability that yield to significant reuse possibilities (7; 8; 9). The 20% of the companies hardly ever perform market analysis to effectively determine which domains will be modeled and which reusable components will be developed, these companies reported a strong and substantial reuse success of 53%, companies that sometimes or normally do (35%) reported a strong and substantial reuse success of 64%, and companies that usually or always do a market analysis (45%) reported a strong and substantial reuse success of 85%, which indicates that market analysis is important to software reuse success.

This market analysis is reflected in the specification of the system and later on the architecture design, the 50% of the companies that always or usually captured, specified, analyzed, and reviewed the requirements taking into account the differences between various products reported a strong and substantial reuse success greater than 74%, companies that hardly ever do (12%) reported a strong and substantial reuse success of 34%. The 82% of the companies that always, usually or normally design an architecture considered the differences between several products reported a strong and substantial reuse success greater than 72%, companies that hardly ever do reported a level of 37%.

According to these results and regression test results, a domain analysis's implementation influences the software reuse success, answering research question **RQ04**.

#### 1.2. RQ2- How is software reuse promoted in companies (organizational factors)?

This section presents an analysis of how the organizational factors influence software reuse success. This analysis is made based on the survey questions' responses, related to research questions, related to the organizational factors.

##### 1.2.1. Software organization and team size

According to the data collected, the following results were obtained:

- *Software organization*: participants tend to work in Startups (see Fig. 3), Therefore, there are representatives from companies of all sizes (the size classification of the companies are based on the statute of the congress of Colombia of 2004 (10)), that allows observing that reuse success level is similar for all sizes of software organizations and all organization sizes claimed a great strong and substantial reuse success, between 70% and 78%, except for micro-enterprise and startups where the level was of 68% and 61% respectively.
- *Team sizes*: all organizations sizes have work teams in software projects of all sizes, except for large organizations where the minimum team size is four people. Software teams in Colombia are mainly made up of three to five people.
- *Project sizes*: all organization sizes work with projects of different sizes, especially with medium-sized projects where the resources (staff, time, supplies, budget, knowledge, etc.) are owned, but some must be adjusted to the projects. Large companies are more into big projects where the resources required to execute a project are sometimes higher than their average capacity. In all projects, sizes were reported a strong and substantial reuse success between 72% and 78%, except for small projects with a strong and substantial reuse level of

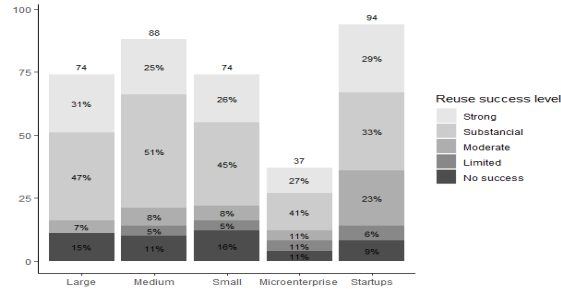


Figure 3: Organization size and reuse success

61%. This non-significant difference indicates that project size does not influence software reuse success.

The regression test result concluded that software team sizes do not influence software reuse success. It concludes that software organization sizes that have the most influence on the software reuse success are Startups, micro-enterprise, small, and medium organizations sizes, answering research question **RQ05**.

#### 1.2.2. Team experience

On the survey review with industry experts, it was found that in Colombia, a person's experience is mainly categorized by two levels, *junior* and *senior*. Junior is a person inexperienced or with little experience, and senior is a person with experience and the ability to help and guide others. Besides that, a company can be conformed by teams of different experiences according to the needs. Each team can have persons with varied experience; for this reason, Fig. ?? shows the total selections by experience level. According to the data collected, the following results were obtained:

- It is identified that most software organizations have at least a team with more seniors than juniors.
- Reuse success levels are similar for all team experience levels, some reported a strong and substantial reuse success between 70% and 77%, except for teams with more juniors than seniors with a strong and substantial reuse success level of 55% where there appears to be less software reuse success.

The results indicate that are team experiences that affect more software reuse success. The regression test result confirms it because it concludes where participants of a team are "all senior", "more senior than junior", and "all junior" have the most influence on the software reuse success.

#### 1.2.3. Legal issues

Legal issues as contracting rules, ownership, and liability are thought to be an impediment to reuse (11; 12). We asked participants under what circumstances the reuse of software in the organization would depend on legal situations. According to the data collected, the following results were obtained:

- The 10% of the organizations claimed that legislation was an impediment, especially when working with software for banking, financial, government and health domains.
- The 28% of organizations claimed that contractual issues are also an impediment due to customer rights because customers request an exclusive or confidentiality agreement and assets cannot be reused.
- The 8% of the organizations indicated that company issues as restrictions and license expiration are impediments.

The regression test result concluded that legal issues should be considered. The types that have the most influence on the software reuse success are legislation, and contractual issues, answering research question **RQ07**.

#### *1.2.4. Economic feasibility*

While reuse promises reducing delivery time-to-market and a cost reduction, it requires an investment in both time and resources to deliver these benefits (13). Participants were asked about the economic feasibility of software reuse in his company .

The majority of the participants agree and strongly agree that software reuse is economically feasible in his company. These participants reported a strong and substantial reuse success level between 72% and 82%. The remaining reported a strong and substantial reuse success level of 50%. Indicating that believing that software reuse is economically feasible affects software reuse success, and according to the regression test result, it seems that economic feasibility does indeed influences software reuse success, answering research question **RQ08**.

#### *1.2.5. Reward and incentives*

Incentives can motivate developers to use reuse strategies. Because of this, participants were asked if the company has a recognition policy (time, money, etc.) to promote software reuse . According to the data collected, the following results were obtained:

- The 62% of participants' responses were agree and strongly agree, these participants reported a strong and substantial reuse success level between 82% and 85%, participants that disagree and strongly disagree reported a reuse success level less than 46%. Indicating that having a recognition policy affects software reuse success.
- The incentives for software reuse are presented mostly in large companies (78% agree/strongly agree to have incentives), unlike the others, the average was 56%. This may be due to the economic capacity of large companies.

According to these results and regression test results, it seems that having organizational incentives does indeed influences software reuse success, answering research question **RQ09**.

#### *1.2.6. Management commitment*

Top management support is necessary to introduce a reuse implementation initiative (14). Given the sentence "The company's top management supports and commits to introduce and maintain software reuse". Following results were obtained:

- It is identified that software reuse is always or usually promoted in all organizations sizes.

- Companies where top management is always involved claimed a strong and substantial reuse success of 92%, in companies where top management is hardly ever involved reported a strong and substantial reuse success of 26%. Indicating that top management commitment affects software reuse success.
- Top Management commitment is presented more in Small and Startups companies.

According to these results and regression test results, it seems that the commitment of top management does indeed influences software reuse success, answering research question **RQ10**.

#### *1.2.7. Software reuse education*

Systematic reuse requires a training effort to understand and undertake reuse concepts. Participants were asked about training and motivation promotion in software reuse in his company.

The 53% of participants' responses were always and usually, these participants reported a strong and substantial reuse success level between 88% and 92%, companies that hardly ever promote training and motivation in software reuse reported a reuse success level of 26%. Indicating that companies that promote training and motivation in software reuse have more software reuse success.

According to these results and regression test results, it seems that training and motivation does indeed influences software reuse success, answering research question **RQ11**.

#### *1.2.8. Independent reusable assets development team*

Having an independent development team to develop reusable assets is a signal of a good reuse process; because these teams are specialized in software reuse, they will deliver certified assets that all teams can reuse, which leads to a centralized repository of assets for reuse that everyone is aware of. According to the data collected, the following results were obtained:

- This factor is not relevant in startup organizations, which usually have less organizational structure (roles and responsibilities) than large organizations.
- The 22% (82 of 367) of the companies indicated to have an independent team for the development of reusable assets, the 87% of these companies reported a strong and substantial reuse success level.
- The 52% (189 of 367) of the companies indicated not have an independent team or role, the 60% of these companies reported a strong and substantial reuse success level. The remaining 26% of the companies are analyzed in subsection 1.3.10.

The results indicate that there is a slight difference between having or not having a team dedicated to developing reusable assets, and regression test result confirms it because concludes that having an independent team that develops reusable software assets do not influence software reuse success, answering research question **RQ12**.

### *1.3. RQ3- How is software reuse applied and controlled (processes factors)?*

This section presents an analysis of how the process factors influence the software reuse success. This analysis is made based on the survey questions' responses, related to research questions, related to the process factors.



#### *1.3.1. Quality models usage*

Quality and maturity models are used to define and evaluate the organization's process to assure the quality of the final product (15; 11). On the validation of the survey, we found companies that use quality models as reference models to improve their processes. Then, in the survey, we asked what quality models are used as a reference or the company is certified. According to the data collected, the following results were obtained:

- The 47% of the companies claimed not to use or possess any quality model, mainly startups and micro-enterprise companies; these companies reported a strong and substantial reuse success of 60%.
- The remaining companies (53%) claimed to use quality models as ISO 15504, CMMI1-CMMI5, and others such as ISO 29110, ISO 9000, ISO 27000, Lean startup, and IT Mark, these companies reported a strong and substantial reuse success between 73% and 87%.
- The quality model with the best reuse success levels (strong and substantial level) were CMMI-4 with 87%, ISO 15504 with 85%, and CMMI-3 with 81%. The quality model with the lowest reuse success level were CMMI-5 with 73% and CMMI-2 with 74%.

The results indicate that are quality models that affect more software reuse success and, regression test result concluded that the quality models that have the most influence on the software reuse success are CMMI-1 to CMMI-4, and ISO 15504, answering research question **RQ13**.

#### *1.3.2. Systematic reuse process*

It is important to have a reuse process to govern the creation of reusable assets to avoid leaving the developers to alter existing assets without any control. Participants were asked if the company has established a software reuse process that software engineers follow .

The companies that agree and strongly agree to follow a software reuse process reported a strong reuse success level of 58% and a substantial reuse success level of 34%, unlike companies that strongly disagree. These claimed not to have strong reuse success and reported a substantial reuse success of 17%.

According to these results and regression test results, it seems that adopting a systematic software reuse process does not influence software reuse success, answering research question **RQ14**. This result differs from what was stated by (15; 1; 16; 17) which indicates that in the Colombian case, it is different, but more future studies are necessary in this case.

#### *1.3.3. Software reuse measurement*

The lack of quantitative measures to assess the benefits and costs is another motive for management unwillingness to adopt reuse strategies (13). Organizations must be able to measure their progress in software reuse, analyze quality and productivity payoff, identify the level of reuse in the assets over time, and identify problems at a time. Participants were asked if the company measures the level of software reuse . According to the data collected, the following results were obtained:

- Large and medium companies measure the most software reuse.
- Companies that strongly agree with the sentence claimed a strong reuse success of 60% and a substantial reuse success level of 34%.

- Companies that strongly disagree claimed a strong reuse success of 14% and a substantial reuse success of 32%.

According to these results and regression test results, the measure of software reuse influences software reuse success, answering research question **RQ15**. This result differs from what was stated by (1) which indicates that in the Colombian case, it is different.

#### *1.3.4. Software certification process*

A process for certifying reused software components allows reusable software components to be highly reliable through operational use. Components should meet specific quality metrics and may be certified as having reached the required quality standards. The reuse of these certified components provides for fault reduction (4). Participants were asked about the effectiveness of the certification process in his company. According to the data collected, the following results were obtained:

- Large and Medium companies follow the most a certification process.
- Companies that strongly agree to have a valuable certification process reported a strong reuse success of 62% and a substantial reuse success level of 26%.
- Companies that strongly disagree claimed a strong reuse success of 12% and a substantial reuse success of 28%.

The results indicate that having an effective software certification process affects more software reuse success. The regression test result confirms it because it concludes that this influences software reuse success, answering research question **RQ16**. This result differs from what was stated by (16) which indicates that in the Colombian case, it is different.

#### *1.3.5. Development of assets for reuse*

The assets generated in the software development life cycle can be created to be reusable in organizations. Table 3 lists the main assets that could be generated in the software development life cycle. Column “Created” shows the total of companies that create the asset, column “Created for reuse” shows the total of companies that develop the asset reusable for use in future applications. In the “other” category, companies indicated that they create mock-ups, prototypes, and process diagrams. According to the data collected, the following results were obtained:

- Results demonstrate that source code is created for reuse and other kinds of assets as requirements, user documentation, architecture, user stories, test cases, test plans, etc.
- The 5% of the software companies claimed not to create reusable assets.
- As to the success in software reuse, the distribution were very similar for all assets, companies claimed a strong success and substantial reuse success between 70% and 80% for each asset, unlike those that don’t create reusable assets reported a success level of 30%, which indicates that create any asset for reuse benefits the company.

After identifying the created reusable assets, we asked participants how often these reusable assets are used; Fig. 4 shows the results. It seems that reusable assets are frequently used, and the creation of these assets influence software reuse success. Thus, the reusable assets are being

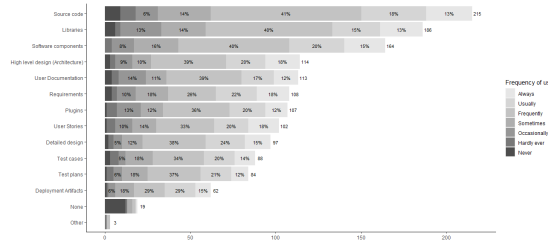


Figure 4: Frequency of use of reusable assets

used and not created just to fulfill a task in the reuse process. The companies that never and hardly ever use the reusable assets (12% of the companies) specified that it is due mainly to the diversity of the projects, ignorance of their existence, individualism between coworkers, it may take a long time when the reusable asset is needed again and a fault of a repository with good documentation.

The results indicate that are assets that affect more software reuse success, and regression test result concluded that the types of developed assets for reuse that have the most influence on the software reuse success are user stories and test plans, answering research question **RQ17**.

#### 1.3.6. Configuration management of the reusable assets

Software reuse requires a certain level of management maturity at different levels regarding application development management and configuration management of the different products involved in the development life cycle. These practices are not always well implemented in organizations with an adequate level of rewarding discipline (14). Participants were asked what assets are under configuration management, table 3, column "Configuration management" shows the results. According to the data collected, the following results were obtained:

- Configuration management is widely used, especially in source code, software components, libraries, and requirements.
- Participants that use configuration management in the reusable assets reported strong and substantial reuse success between 70% and 79% for all assets, excluding the "none" and "other" categories.
- Companies that claimed not to use configuration management reported a strong and substantial reuse success of 36%.

The results indicate that are configuration management assets that affect more software reuse success, and regression test result confirms it because it concludes that the configuration management and change control assets that have the most influence on the software reuse success are deployment artifacts, user documentation, and detailed design, answering research question **RQ18**.

#### 1.3.7. Reused assets type identification

It is identified that companies reuse source code and other assets created in different phases of a software development life cycle. According to the data collected, the following results were obtained:

Table 3: Assets for reuse

Asset	Created	Created for reuse	Configuration management
Source code	77%	59%	57%
Requirements	57%	29%	36%
User documentation	56%	31%	29%
High level design (architecture)	55%	31%	27%
Software components	54%	45%	37%
User stories	53%	28%	34%
Test cases	51%	24%	22%
Detailed design	46%	26%	29%
Libraries	45%	45%	38%
Test plans	43%	23%	22%
Plugins	32%	29%	24%
Deployment artifacts	28%	17%	18%
Other	3%	1%	-
None	-	5%	7%

- Companies that do not reuse assets such as requirements, user histories, architecture, detailed design, and test cases reported a strong and substantial reuse success level between 55% and 60%.
- Companies that do not reuse software components reported a strong and substantial reuse success level of 40%.
- Companies that reuse any of the listed types of assets reported a strong and substantial reuse success level greater than 77%.
- Only 4% of the participants reported reusing all listed types of assets completely and obtained substantial and strong reuse success for all assets.

The results indicate that are reused assets that affect more software reuse success. The regression test result confirms it because it concludes that the types of reused assets for reuse that have the most influence on the software reuse success are requirements, user histories, high-level design (architecture), and software components, answering research question **RQ19**.

#### 1.3.8. Previous development of reusable assets

At the beginning of a software development project, reuse events should be planned (8; 11). According to the data collected, the following results were obtained:

- The 21% of the participants plan which assets should be created at the beginning of a project. The 57% of the participants make the assets just at the moment that are needed. The 10% use both techniques, and 12% of the participants indicated not to develop reusable assets.
- Create the assets just when needed is a suitable technique when combined with a systematic technique. As a result, companies that plan before or use both techniques (plan before and create the assets at the moment) reported a strong and substantial reuse success of 82%.

- Companies that only create the assets at the moment reported a level of 76%, and the companies that do not develop reusable assets reported a strong and substantial reuse success of 21%.

These results indicate that software reuse is usually done opportunistically; if during the software development life cycle, members of the development team see the possibility of reusing some asset, they do it, but that does not happen systematically. Using any development techniques for the development of reusable assets has a good impact on software reuse success. Regression test result confirms it concluding that having a reusable assets' pre-development does not influence software reuse success, answering research question **RQ20**. This result differs from what was stated by (15) which indicates that in the Colombian case, it is different.

#### *1.3.9. Origin of the reused assets*

The companies must know the source of their reused software assets. These assets may be developed internally or externally. For example, develop reusable assets from scratch, from existing products, adapt them, or modify them to fit new products. According to the data collected, the following results were obtained:

- The 64% of the participants indicated to create reusable assets from scratch, 63% take assets from existing projects and improve it, 31% made re-engineering of existing products in the market, 24% take assets from existing projects with no modification and 12% use COTS.
- Companies that use combined techniques reported a strong and substantial reuse success between 60% and 89%, where the highest reuse success level was obtained creating assets from scratch, enhancing existing assets, and re-engineering assets in the market. The lowest reuse success level was for the assets created from scratch and the reuse of exact copies or existing projects.
- Companies that create reusable assets only from scratch reported a strong and substantial reuse success level of 43%.

These results indicate that having different types of sources of the reusable assets have more impact on software reuse success, even if assets are developed from scratch. The regression test result concluded that types of origins of reusable assets that have the most influence on the software reuse success are when "assets are enhanced copies of existing work", "assets are developed from the re-engineering of existing products, not necessarily their own products", and "assets are unmodified existing project work", answering research question **RQ21**.

#### *1.3.10. Specific function in the software reuse process*

In subsection 1.2.8 it was identified that having a team dedicated to developing reusable assets influences software reuse. This section identifies if having a role (not a team) influences software reuse success. According to the data collected, the following results were obtained:

- The 52% (189 of 367) of the participants indicated not to have a team or role dedicated to software reuse; these companies reported a strong and substantial reuse success of 60%.
- The 12% (44 of 367) of the participants contract a third party to develop the reusable assets; these companies reported a strong and substantial reuse success of 75%.

- The 26% (86 of 367) of the companies have a role dedicated to software reuse processes; these companies reported a strong and substantial reuse success of 84%.
- Most companies that do not have a specific team or role for software reuse and achieved a strong and substantial reuse success indicated that all members contribute to creating a library of reusable assets. It is everyone's job to analyze similar components and search for new solutions. This is presented more in startups, micro-enterprises, and small companies, especially in Startups where 74% of the companies do not have a role or team dedicated to software reuse due to its internal capacity.

According to these results and regression test results, having a specific role for the software reuse process does not influence software reuse success, answering research question **RQ22**. This confirms the result present by (17).

#### 1.4. RQ4- How is software reuse supported (technological factors)?

This section presents an analysis of how the technological factors influence software reuse success. This analysis is made based on the survey questions' responses, related to research questions, related to the technological factors.

##### 1.4.1. Software development approach

To identify the influence of the software development approach on software reuse success, we asked which methods, methodologies, practices, and programming paradigms are used in the company.

**Models used in software development life cycle.** The most used models are the Waterfall model with 27% and Iterative and incremental development model with 21%, Followed by Spiral model 14% and V-Model 10%.

**Software development methodologies.** The most used methodology is Scrum with 54% of the companies, the 28% of the participants use their methodology, the remaining use methodologies as Kanban — 14%, Extreme programming (XP) — 13%, Feature-driven development (FDD) — 7%, Enterprise Agile (previously XBreed) — 3%, Agile Unified Process (AUP) — 3% and CRYSTAL — 3.5%.

**Software development process and practices.** The most used are Test-driven development (TDD) — 19%, Unified process (UP) — 7%, Adaptive Software Development (ASD) — 2%. 16% of the participants use DevOps practices.

**Programming paradigms.** Most of the companies use object-oriented programming (81%), followed by component-oriented programming (35%) and functional programming (20%) as presented in Fig. 5.

For the programming paradigms, companies reported a strong and substantial reuse success between 70% and 86%, where the lowest level was for object-oriented programming and the highest for constraint-based programming. The regression test result concluded that the software development approaches that influence software reuse success are when companies use FDD, their methodology, Scrum, aspect-oriented programming, procedural programming, constraint-based programming, and reactive programming, answering research question **RQ23**.

#### 1.4.2. Programming language

To identify the influence of programming language on reuse success, we asked participants to select the company's programming languages for back-end and front-end in software projects. Fig. 6 shows the results; in the "other" category, the following languages are grouped: Scala, SAP ABAP, TypeScript, Go, Kotlin, Clojure, Delphi, Pascal, Elixir, Erlang, INFORMIX-4GL, PL/SQL, Ruby, Swift, and Visual FoxPro.

It is worth noting that the most used programming language is Javascript, followed by SQL, Java, and PHP. The programming language with the highest strong and substantial reuse success level was C with 86% and C++ with 77%; the lowest were Objective-C with 69% and C# with 67%. The most used front-end framework is Angular, followed by Bootstrap, React, Vue.js, JQuery, Ionic. For back-end is ASP.NET, followed by Lavarel, Spring, Django, Node.js, Express.js, CodeIgniter, and Yii.

The regression test result concluded that the programming languages that have the most influence on the software reuse success are C#, C++ and SQL, answering research question **RQ24**.

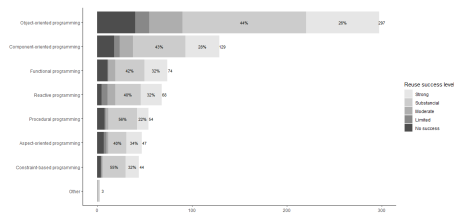


Figure 5: Programming paradigms and reuse success

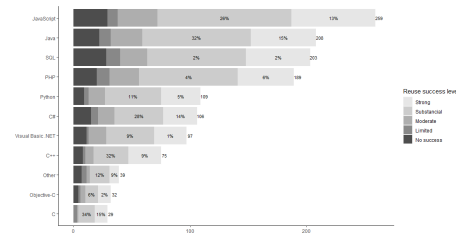


Figure 6: Programming languages and reuse success

#### 1.4.3. Repository systems usage

A reuse repository is a collection of reusable assets with requirements such as search and retrieval mechanisms (18), allowing teams to determine if the desired reusable asset is available. As repositories are widely used, especially for source code. Participants were asked about the effectiveness and efficiency of the repository in his company. Companies that strongly agree to have an effective and efficient repository reported a strong and substantial reuse success of 86%, neutral companies reported a level of 42%, and companies that strongly disagree reported a level of 40%.

The results indicate that having an effective repository system affects software reuse success. The regression test result confirms it because it concludes that repositories influence software reuse success, answering research question **RQ25**.

#### 1.4.4. CASE Tools usage

The tools used by organizations should support the software reuse process, Participants were asked about the tools used to promote software reuse in his company. According to the data collected, the following results were obtained:

- The majority agree that the used tools in the company promote software reuse in projects; these companies reported a strong and substantial reuse success of 73%, companies that disagree reported a level of 31%.

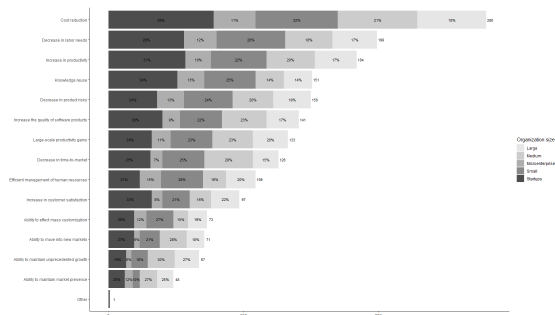


Figure 7: Expectations in software reuse practices

- The tools used by the participants to support software reuse are the repositories for code hosting, version control, and collaboration such as Github, Bitbucket, Maven repository, and Apache Subversion.
- Package managers are also widely used; some participants use copy and paste as a reuse technique.

The results indicate that the use of CASE tools affects software reuse success. The regression test result confirms it because it concludes the this influences software reuse success, answering research question **RQ26**.

#### 1.5. RQ5- What are the expectations and adoption barriers in the context characteristics (organization size)?

Fig. 7 and 8 shows the expectations and adoption barriers that organizations have actually in software reuse practices.

Fig. 7 shows that the participants considered cost reduction, decrease in labor needs and increase in productivity as the most important expectations when using software reuse practices. By organization size, all agree with that expectations. The reuse of knowledge is considered important due to the advantages it brings to the collaboration of teams; teams will learn of the knowledge in these reusable assets and become the basis for creating new reusable assets or products. Participants indicated that also helps to minimize delivery times with each project.

Fig. 8 shows that the principal barrier for adopting software reuse practices is the diversity of the running projects. This is because most of the participants work in companies dedicated to develop custom software. As second barrier the lack of time and resources, and the third is the lack of communication between employees. In the “other” category, some participants pointed out that the customers’ various restrictions are an adoption barrier and the requests made in the initial phases of software development, which leads to contractual issues associated with the factor of *legal issues*.

The main adoption barrier in large companies, unlike the other ones, was the lack of communication between employees. A possible explanation of this result can be the large number of teams dedicated to different projects complicating the communication. We also identified that to overcome this barrier, it becomes essential to have a good repository of well-documented reusable assets with search and retrieval mechanisms.



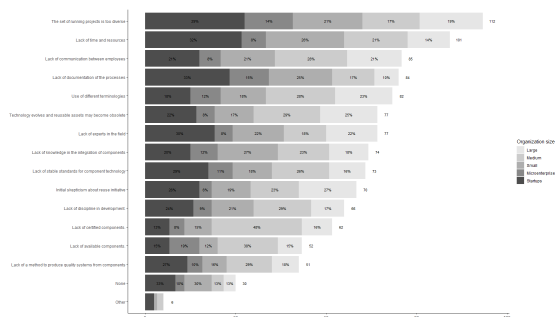


Figure 8: Adoption barriers in software reuse practices

## References

- [1] W. B. Frakes and C. J. Fox, "Sixteen questions about software reuse," *Communications of the ACM*, vol. 38, no. 6, pp. 75–ff., jun 1995. [Online]. Available: <http://doi.acm.org/10.1145/203241.203260> <http://portal.acm.org/citation.cfm?doid=203241.203260>
- [2] P. Clements and L. Northrop, *Software product lines : practices and patterns*. USA: Addison-Wesley, 2002. [Online]. Available: [https://books.google.com.co/books/about/Software\\_Product\\_Lines.html?id=tHGFQgAACAAJ&redir\\_esc=y](https://books.google.com.co/books/about/Software_Product_Lines.html?id=tHGFQgAACAAJ&redir_esc=y)
- [3] V. Alves, N. Niu, C. Alves, and G. Valença, "Requirements engineering for software product lines: A systematic literature review," *Information and Software Technology*, vol. 52, no. 8, pp. 806–820, aug 2010. [Online]. Available: <https://ezproxy.eafit.edu.co:2072/science/article/pii/S0950584910000625>
- [4] D. Rine and N. Nada, "An empirical study of a software reuse reference model," *Information and Software Technology*, vol. 42, no. 1, pp. 47–65, jan 2000. [Online]. Available: <https://ezproxy.eafit.edu.co:2072/science/article/pii/S0950584999000555>
- [5] F. van der Linden, K. Schmid, and E. Rommes, *Software Product Lines in Action*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007. [Online]. Available: <http://link.springer.com/10.1007/978-3-540-71437-8>
- [6] M. Jha and L. O'Brien, *Identifying Issues and Concerns in Software Reuse in Software Product Lines*. Berlin, Heidelberg: Springer Berlin Heidelberg, sep 2009, vol. 5791.
- [7] C. Sholom G, "Product Line State of the Practice Report," 2002. [Online]. Available: <https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=5961>
- [8] W. Spoelstra, M. Iacob, and M. van Sinderen, "Software reuse in agile development organizations: a conceptual management tool," in *Proceedings of the 2011 ACM Symposium on Applied Computing - SAC '11*. New York, New York, USA: ACM Press, 2011, p. 315. [Online]. Available: <http://portal.acm.org/citation.cfm?doid=1982185.1982255>
- [9] R. J. Leach, *Software Reuse, Second Edition: Methods, Models, Costs*, 2nd ed. New York, NY: McGraw-Hill New York, 2012. [Online]. Available: <https://www.amazon.com/Software-Reuse-Second-Methods-Models/dp/1939142350>
- [10] El congreso de Colombia, "Ley 905 de 2004," pp. 1–2, 2004. [Online]. Available: <http://www.mipymes.gov.co/mipymes/media/mipymes/Documentos/LEY-905-DE-AGOSTO-2-DE-2004.pdf>
- [11] V. Garcia, D. Lucrédio, A. Alvaro, E. Santana De Almeida, R. Fortes, M. Fortes, S. Romero, and L. Meira, "Towards a Maturity Model for a Reuse Incremental Adoption," Institute of Mathematical and Computing Sciences, Sao Paulo, Brazil, Tech. Rep., oct 2007.
- [12] W. Lim, "Legal and contractual issues in software reuse," in *Proceedings of Fourth IEEE International Conference on Software Reuse*. USA: IEEE Comput. Soc. Press, 1996, pp. 156–164. [Online]. Available: <http://ieeexplore.ieee.org/document/496123/>
- [13] K. Sherif and A. Vinze, "Barriers to adoption of software reuse: A qualitative study," *Information & Management*, vol. 41, no. 2, pp. 159–175, dec 2003. [Online]. Available: <https://www.sciencedirect-com.ezproxy.eafit.edu.co/science/article/pii/S0378720603000454>
- [14] T. Käkölä and J. C. Duenas, Eds., *Software Product Lines*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. [Online]. Available: <http://link.springer.com/10.1007/978-3-540-33253-4>
- [15] D. Lucrédio, K. dos Santos Brito, A. Alvaro, V. C. Garcia, E. S. de Almeida, R. P. de Mattos Fortes, and S. L. Meira,

- “Software reuse: The Brazilian industry scenario,” *Journal of Systems and Software*, vol. 81, no. 6, pp. 996–1013, jun 2008. [Online]. Available: <https://ezproxy.eafit.edu.co:2072/science/article/pii/S0164121207002221>
- [16] D. C. Rine, “Success Factors for Software Reuse That Are Applicable Across Domains and Businesses,” in *Proceedings of the 1997 ACM Symposium on Applied Computing*, ser. SAC '97. New York, NY, USA: ACM, 1997, pp. 182–186. [Online]. Available: <http://doi.acm.org/10.1145/331697.331736>
  - [17] M. Morisio, M. Ezran, and C. Tully, “Success and failure factors in software reuse,” *IEEE Transactions on Software Engineering*, vol. 28, no. 4, pp. 340–357, apr 2002.
  - [18] A. Mili, R. Mili, and R. T. Mittermeir, “A survey of software reuse libraries,” *Ann. Softw. Eng.*, vol. 5, no. 1, pp. 349–414, Jan. 1998. [Online]. Available: <http://dl.acm.org/citation.cfm?id=590631.590637>