InsighTD Survey
Investigating causes and implications of Technical Debt

List of TD causes and their categories identified in InsighTD BR and USA

The categories identified were:

- **Lack of knowledge:** refers to the causes related to the team's lack of knowledge to develop the project. This category grouped six causes, such as *lack of understanding* and *lack of experience*;
- **External factors:** brings together causes that are related to issues that are external to the development team and organization. With 13 causes, this category grouped items such as *discontinued component* and *customers do not know their own needs*;
- **Infrastructure:** refers to causes related to tools, technologies, and development environments. With five causes, this category groups causes such as *inadequate use of tools* and *unavailable infrastructure*;
- **Methodology:** groups causes related to processes and methodologies used in the development of the project. There were 18 causes associated with this category, which includes items like *lack of a well defined process* and *lack of code review*;
- **Organizational:** groups organization level related causes. This category groups items such as *inadequate management decision* and *lack of qualified professionals*, totaling eight causes;
- **People:** groups causes directly related to members of software development teams. This category grouped 11 causes like *lack of commitment* and *overload of the team*;
- **Planning and management:** groups causes related to project planning and management. In this category, we have 17 as, for example, *inappropriate planning* and *poor resource allocation*;
- **Development issues:** groups issues that occur during project development. With 22 associated causes, this category has items like *change in design* and *non-adoption of good practices*.

List of causes:

Development Issues

Adoption of contour solutions as definitive
Avoid changes in working code
Sloppy code
Change in design
Change of scope
Complexity of the project
Inaccurate or complex requirement
Inadequate data model
Inadequate technical decision
Lack of quality
Legacy system
No bug fixes
Change of requirements
Non-adoption of good practices
Non-compliance with non-functional requirements
Poor choice of framework
Poor scope definition
Poor design

Problems in architecture
Requirement definition issues
Requirements elicitation issues
Version Incompatibility

External factors

Change in external component
Customer does not know his own needs
Customer does not listen the project team
Customer does not willing to pay
Discontinued component
External component dependency
External component limitation
Lack of confidence in the product
Politics in the business
Pressure
Structural change in the involved organizations
Third party team involved in the project
Updating existing tools

Infrastructure

Inadequate choice of technology / tool / platform
Inappropriate use of tools
Integration of new tools
Problems with test environment
Required infrastructure unavailable

Lack of knowledge

Lack of technical knowledge
Lack of experience
Lack of knowledge on development tools
Lack of domain knowledge
Lack of understanding
Lack of information

Methodology

Inappropriate / poorly planned / poorly executed test
Lack of a well-defined process
Lack of automated testing
Lack of change control
Lack of code review
Lack of IT Governance
Lack of pair programming
Lack of prototyping
Lack of refactoring
Lack of requirements analysis

Lack of reuse practices

Lack of traceability of bugs

Lack of validation

Non-compliance with policies established by management

Nonexistent documentation

Outdated / incomplete documentation

Over documentation

Test not performed

Organizational

High turnover of the team

Inadequate management decision

Lack of prioritization at company level

Lack of priority for the project

Lack of qualified professionals

Lack of specific team

Lack of training

The company does not give importance to documentation

People

Be responsable for code from others

Developers do not like to do some activities

Lack of commitment

Lack of confidence in the development team

Lack of interest in acquiring knowledge

Lack of motivation

Lack of team communication

Lack of transparency between customer and development team

Low productivity

Non-sharing of knowledge

Team overload

Planning and management

Concern with just back-end development

Cost

Deadline

Do not recognize the impact of TD in software projects

Focus on producing more at the expense of quality

Inaccurate time estimate

Inadequate impact and risk analysis

Inappropriate planning

Insufficient long-term vision

Lack of perception of the importance of automated tests

Lack of perception of the importance of dealing with TD

Lack of perception of the importance of dealing with TD
Lack of perception of the importance of testing and refactoring
Manager's lack of awareness of customer needs
Not effective project management
Poor allocation of resources
Poorly elaborated SLAs