**List of Preventive Actions for TD and their Categories Identified in InsighTD BR and USA**

**The categories identified were:**

* **Adopting of good practices for software development:** includes the actions related to software development activities. With 31 preventive actions, this category grouped items such as *adoption of good practices* and *well-defined documentation*;
* **Controlling and measuring the quality in the project:** groups actions associated to quality assurance.There were 28 preventive actions associated with this category, which includes items like *creating tests* and *code evaluation*;
* **Controlling and negotiating the software requirements:** includes actions related to requirements engineering activities. This category groups items such as *well-defined requirement* and *good communication between stakeholders*, totaling nine preventive actions;
* **Following and improving a well-defined process:** organizes the actions associated to process and its management. This category grouped nine preventive actions like *following well-defined project process* and *iterative process*;
* **Following a well-defined project planning:** is related to the actions associated with project planning activities**.** This category grouped 21 preventive actions, such as *following the project planning* and *risk and impact analysis*;
* **Having an effective team:** encompasses the actions that improve the technical knowledge and the motivation of the team. With 26 preventive actions, this category groups preventive actions such as *training* and *good communication on team*;
* **Identifying, managing, and estimating TD:** contains the actions applied to the TD management. In this category, we have 12 as, for example, *implementation of a TD payment strategy* and *TD monitoring*.

**List of preventive actions for TD:**

|  |
| --- |
| **Adopting of good practices for software development** |
| Adoption of architectural pattern |
| Adoption of good practices |
| Appropriate use of design pattern |
| Available documentation |
| Bug fixing |
| Bug tracking |
| Considering technical constraints |
| Database administrator participation in other development tasks |
| Discarding prototypes |
| Documentation update |
| Evaluation of the prototypes |
| Following architectural pattern |
| Following well-defined development standards |
| Improving the software architecture |
| Non-using reuse in all situations |
| Project design |
| Providing design and requirements sooner |
| Reusing code |
| Spend more time on the design and implementation of UI |
| Upfront design of the project |
| Usability design |
| Use the most appropriate version of the technology |
| Using a development framework |
| Using agile practices |
| Using Components |
| Using continuous integration |
| Using good design practices |
| Version control |
| Well-defined architecture |
| Well-defined documentation |
| Well-defined ER model |
| **Controlling and measuring the quality in the project** |
| Adoption of pair programming |
| Appropriate test coverage |
| Avoiding coding with coupling |
| Code and Architecture review |
| Code evaluation |
| Code standardization |
| Continuous revision of plans |
| Creating automated tests |
| Creating tests |
| Eliminating code dependencies |
| Framework update |
| Improving the maintainability of the project |
| Organizing code repository |
| Planning code structure |
| Planning maintenance strategies |
| Quality as priority |
| Quality control |
| Refactoring |
| Review of artifacts |
| Tools for checking good practices of design and code |
| Using acceptance level testing |
| Using flexible code |
| Using integration tests |
| Using regression testing |
| Validating requirement |
| Validating tests |
| Verifying and validating activities |
| Well-defined metrics |
| **Controlling and negotiating the software requirements** |
| Awareness of the impact of business decisions on technology |
| Considering the requirements in all activities |
| Customer commitment |
| Focus on end requirements |
| Good communication between stakeholders |
| Identifying the core requirement |
| Requirements changes tracking |
| Well-defined requirement |
| Well-defined scope statement |
| **Following and improving a well-defined process** |
| Better understanding of development process by businesses |
| Flexibilization in the defined process |
| Focusing on agile delivery |
| Following well-defined project process |
| Improving software development process |
| Iterative process |
| Monitoring of the process |
| Understanding development process followed by team |
| Using collaborative process |
| **Following a well-defined project planning** |
| Adding time to estimate tasks |
| Allocation of qualified professionals |
| Appropriate tasks allocation |
| Better Project Management |
| Changing team priorities slowly |
| Controlling changes on project |
| Effective monitoring |
| Flexibility in deadlines |
| Focusing on long-term goals |
| Following the project planning |
| Improve the understanding of TD concept by PMs |
| Internal communication |
| Organizational support |
| Participation of stakeholders in planning activities |
| Project manager's participation |
| Risk and impact analysis |
| Solve problem at identification |
| Standardization in carrying out activities |
| Task automatization |
| Well planned deadlines |
| Well-defined effort estimation methods |
| **Having an effective team** |
| Adequate technical management |
| Assertive actions |
| Being committed |
| Being open to learn new things |
| Being proactive to TD payment |
| Contracting a domain expert to architect the project |
| Discipline |
| Discussing about improvements on project |
| Do it right in the first time |
| Focus |
| Good allocation of resources in the team |
| Good communication on team |
| Good group work |
| Having an emotional stability team |
| Having expert consultation on design choices |
| Motivate good team work |
| Organized team |
| Readiness of team |
| Team open to changes |
| Technical knowledge |
| Technical leader working together with the team |
| Training |
| Training on Refactoring |
| Training on reviewing code |
| Training on TD |
| Understanding the technology in use |
| **Identifying, managing, and estimating TD** |
| Historical knowledge on TD |
| Implementation of a TD identification strategy |
| Implementation of a TD management strategy |
| Implementation of a TD payment strategy |
| Informing about extra effort due TD |
| Prioritization of TD payment |
| Raising awareness of the debt |
| TD mitigation |
| TD monitoring |
| TD repayment effort |
| TD tracking |
| Use of tools to identify TD |