**17 TD Decision Rules**

|  |  |  |  |
| --- | --- | --- | --- |
| **Legend** | **TD Accumulation** | **TD Repayment** | **TD Prevention** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **#** | **Decision rules** | **Decision factors** | **Time factor** | **Decision reference** |
| 1 | IF financial resources are limited OR required time to market is tight OR the product uncertainty is high, THEN a startup team should consider an architecture or infrastructure that facilitates faster development | * Financial resources * Required time to market * The product uncertainty | T1.1  T2.1 | S1\_D1  S2\_D1  S1\_D2  S3\_D5  S4\_D4 |
| 2 | IF a startup team is more familiar with a sub-optimal development infrastructure (e.g., development framework or tool), THEN the team should adopt the sub-optimal infrastructure | * Familiarity with the development infrastructure | T1.1 | **S1\_D1**  S2\_D2  S3\_D1  S3\_D2 |
| 3 | IF 3rd party infrastructure/services are available, THEN a startup team should consider using them for the development | * Availability of 3rd party infrastructure or services | T1.1  T2.1 | **S4\_D2**  S4\_D5 |
| 4 | A startup team should enforce some basic coding standards to be followed within the team | **--** | T1.2 | **S2\_D3** |
| 5 | A startup team should enforce minimum unit test for the important functions | **--** | T1.2 | **S1\_D3**  **S2\_D4**  S3\_D4  S4\_D1 |
| 6 | IF financial resources are limited OR required time to market is tight, THEN a startup team should skip the other (unenforced) coding standards | * Financial resources * Required time to market | T1.2 | S2\_D3  S4\_D2 |
| 7 | IF an existing infrastructure/architecture restricts evolution, THEN a startup team should fix/replace the infrastructure/architecture that restricts evolution | * Evolution restriction | T1.2  T2.1 | S3\_D3  **S4\_D2**  S4\_D6  S4\_D7 |
| 8 | A startup team should enforce manual system test within an allocated time | **--** | T1.3 | **S1\_D3**  **S2\_D4**  **S4\_D1** |
| 9 | IF financial resources are limited OR required time to market is tight OR the product uncertainty is high, THEN a startup team should skip the other (unenforced) testing | * Financial resources * Required time to market * The product uncertainty | T1.3 | S1\_D3  S2\_D4  S3\_D4  S4\_D1 |
| 10 | IF the framework upgrade workload is feasible OR the framework upgrade process is automated, THEN a startup team should upgrade the development framework to the latest version | * The framework upgrade workload * The framework upgrade process | T2.1  T2.2 | **S2\_D5**  S4\_D3 |
| 11 | IF financial resources are available, THEN a startup team should gradually replace some scripting with strong typed language | * Financial resources | T2.1  T2.2 | **S3\_D1** |
| 12 | IF financial resources are available OR  bug rate is high, THEN a startup team should include integration test for the important integrations | * Financial resources * Bug rate | T2.2 | **S1\_D3**  S1\_D4  **S3\_D4**  **S4\_D1** |
| 13 | IF the growth of development team is high (or expected to be high soon), THEN a startup team should document the requirements of some important functions | * Team growth | T2.2  T2.3 | **S3\_D4** |
| 14 | IF an existing infrastructure/architecture restricts evolution OR system performance is bad, THEN a startup team should consider restructuring the software | * Evolution * System performance | T2.3 | S1\_D5  S2\_D6 |
| 15 | IF financial resources are available, THEN a startup team should gradually restructure the software | * Financial resources | T2.4 | S2\_D6  S2\_D7  S2\_D8 |
| 16 | IF financial resources are limited, THEN a startup team should freeze the existing code as legacy AND develop the upcoming features on a new development infrastructure. | * Financial resources | T2.4 | S1\_D5 |
| 17 | A startup team should consider test improvement for the new/refactored code. | -- | T2.4 | S1\_D6  S2\_D7  S2\_D9 |