**Analiza sistemului software ”Toolset”**

Istoric document

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Description** | **Date** |
| 1.1 | Costean Adrian | Crearea unei prime analiza a sistemului ”Toolset” si definirea unor cerinte cheie in dezvoltarea acestui sistem | 12.10.2017 |
|  |  |  |  |
|  |  |  |  |

Cuprins

[1 Introducere 4](#_Toc494750775)

[1.1 Obiectiv 4](#_Toc494750776)

[1.2 Scopul 4](#_Toc494750777)

[1.3 Limitari 4](#_Toc494750778)

[2 Cerinte 4](#_Toc494750779)

[2.1 Cerinte functionale 4](#_Toc494750780)

[2.2 Cerinte non-functionale 4](#_Toc494750781)

[2.3 Cerinte care influenteaza arhitectura 5](#_Toc494750782)

[3 Componente 5](#_Toc494750783)

[3.1 Descopmunere in componente 5](#_Toc494750784)

[3.2 Analiza functionala a componentelor 5](#_Toc494750785)

[3.3 Argumentare 5](#_Toc494750786)

[4 Prezentare sistem 5](#_Toc494750787)

[4.1 Perspectiva 1 (cu diagrame) 5](#_Toc494750788)

[4.2 Perspectiva 2 (cu diagrame) 5](#_Toc494750789)

[5 Indicatori de calitate 5](#_Toc494750790)

[5.1 Indicator 1 5](#_Toc494750791)

[5.2 Indicator 2 5](#_Toc494750792)

[5.3 Indicator 3 5](#_Toc494750793)

[6 Tehnologii middleware 6](#_Toc494750794)

[7 Modele si stiluri arhitecturale 6](#_Toc494750795)

[8 Scenarii de validare 6](#_Toc494750796)

# Introducere

## Obiectiv

In cadrul companiei Continental, departamentul IBS ( Interior, Body & Security) ocupa un rol important in dezvolatarea software si hardware a elementelor ce tin de siguranta pasagerilor ( airbag-uri, sisteme de avertizare a soferului, frane). In ultimii 20 de ani asistam la o dezvoltare tehnologica exponentiala, prin urmare si metodele, programele, utilitarele trebuie sa evolueze in acest ritm pentru a asigura o aliniere cu stadardele si concurenta.

## Scopul

Comparand activitatea unui programator de acum 20 ani, cu activitatea unui programator de acum, putem observa foarte bine ca procesele au devenit din ce in ce mai complexe, cerintele care tin de calitatea codului scris, de validarea acestuia, de eficienta s.a.m.d sunt din ce in ce mai mari, de asemenea raportul de timp in care un produs trebuie livrrat e mult mai mic. Tinand cont de aceste constrangeri, e aproape imposibil ca un programator sa combine toate aceste elemente si sa tina evidenta lor, de aceea e nevoie de un utilitar care sa automatizeze pe cat posibil operatiile repetitive, sa adauge functionalitati de verificare a codului, de validare a acestuia si de intocmire a unor analize de performanta si calitate si multe alte elemente care sa asigure o eficientizare si armonizare a procesului de dezvoltare software.

## Limitari

Pentru a evidentia necesitatea implementrii sistemului descirs in acest document, putem porni spre exemplu de la un simplu proiect C/C++. Un dezvoltator trebuie sa ruleze intai manual generatoare de cod, sa compileze manual sursele C asigurandu-se ca adauga toate optiunile de compilator, sa ruleze teste de calitate, sa ruleze un linkeditor care sa asambleze intreg proiectul intr-o librarie finala care va fi incarcata pe componenta hardware, sa isi faca copie de siguranta a proiectului intr-o baza de date etc. Munca in echipa devine destul de dificila, impartirea task-urilor e greu de realizat, nu exista o siguranta si un proces bine definit prin care sa poata fi verificata calitatea produsului final si multe alte limitari care apar in timpul procesului de dezvoltare.

# Cerinte

## Cerinte functionale

### Sisteme de operare

Sistemul de operare suportat de produs trebuie sa fie Windows cu posibilitate de extindere la Linux in viitor.

### Fisier de configurare

Fiecare proiect va trebui sa poata avea un fisier de configurare global unde va putea modifica urmatorii parametrii

* versiuni de tool-uri (interne/externe)
* tipul de compilator
* parametrii pentru compilator sau linker
* fisierele sursa care vor fi incluse/excluse din proiect
* Activarea/dezactivare conexiunii la sistemul de versionare

### Conexiune la sisteme de versionare

Produsul “Toolset” trebuie sa asigure suport pentru conexiunea cu sistemele de versionare Sinergy, IMS, Perforce sau GitHub unde este stocat proiectul C/C++ pentru a asigura o mai buna disciplina.

Conectarea se face cu credentialele userului care ruleaza produsul, iar setarile de conectare trebuie sa poata fii configurabile dintr-un fisierul de configurare (2.1.2) de catre user. O data conectat la sistemul de versionare, produsul trebuie sa ruleze querry-uri pentru a colecta informatiile legate de proiectul care se afla in mediul de lucru.

Aceste informatii trebuie sa cuprinda :

* Versiunea fiecarui fisier prezent in proiect
* Versiunea curenta a proiectului
* Numele real al proiectului
* Rolul userului conectat

### Suport compilatoare

Proiectele sunt concepute in functie de hardware pentru anumite tipuri de microprocesoare, din cauza aceasta, produsul trebuie conceput in asa masura incat userul sa poata configura compilatorul dintr-un fisier de configurare xml. Suportul trebuie asigurat pentru compilatoarele din familia gh, arm si iar dar sa aiba posibilitatea de a fi extins daca apare nevoia aceasta.

### Sistem de dependinte

In cazul unei modificari in proiect, timpul necesar recompilarii intregului proiect trebuie eliminat prin crearea unui sistem de depndinte care sa detecteze care fisiere din proiect sunt afectate de schimbare si sa le marcheze pentru compilare doar pe acelea.

### Linkeditare

Fisierele obiect rezultate in urma compilarii impreuna cu librariile scrise sau generate trebuie sa poata fii asamblate intr-un singur fisier care va fi incarcat in componenta hardware. Pentru aceasta operatie, parametrii de configurare trebuie sa poata fi specificati intr-un fisier xml de configurare.

### Raport de calitate al codului

Produsul trebuie sa ofere posibilitatea de a crea un raport de calitate per proiect. Fiecare fisier inclus in proiect trebuie sa fie analizat din punct de vedere al calitatii codului scris si al standardelor de codare MISRA.

Rezultatele analizei trebuie sa fie generate intr-un raport xlsx si clasificate in functie de tipul fisierelor, tipul erorilor si importanta problemei detectate.

## Cerinte non-functionale

[Describe t]

## Cerinte care influenteaza arhitectura

[Care dintre cele mai sus au impact asupra arhitecturii]

# Componente

## Descopmunere in componente

## Analiza functionala a componentelor

## Argumentare

# Prezentare sistem

## Perspectiva 1 (cu diagrame)

## Perspectiva 2 (cu diagrame)

# Indicatori de calitate

## Indicator 1

## Indicator 2

## Indicator 3

# Tehnologii middleware

# Modele si stiluri arhitecturale

# Scenarii de validare