

7.1. Modele de dezvoltare software – V-Modell

7.1.1. Introducere in conceptul V-modell

Există numeroase abordări pentru dezvoltarea sistemelor, definite și proiectate care sunt folosite în procesul de dezvoltare sistemelor electronice. *Modelul V* (V Model) este un asemenea model, proiectat să faciliteze înțelegerea procesul complex asociat cu dezvoltarea sistemelor. In ingineria sistemelor este folosit pentru a defini un proces uniform pentru dezvoltarea produselor sau a proiectelor.

În general, un sistem de calitate livrat către utilizator, trebuie să:

- Îndeplinească sau să depășească cerințele utilizatorului/clientului
- Să se încadreze în limitele de cost prestabilite
- Să lucreze în mod eficient în infrastructura pentru care a fost proiectat

7.1. Modele de dezvoltare software – V-Model

7.1.2. Prezentare generala

Modelul V a apărut din necesitatea de a începe testarea încă din faza de proiectare a sistemelor. Acesta recomandă ca atât dezvoltarea sistemului cât și testarea acestuia să înceapă din același punct, referindu-se la aceleași informații. Acest fapt se datorează considerentelor economice, dat fiind faptul că cu cât o eroare în sistem este descoperită mai târziu, cu atât costul de corecție al acestuia crește. Estimările spun că un defect descoperit în timpul proiectării costă o unitate monetară pentru a fi corectat, același defect găsit chiar înainte de procesul de testare 6.5 unități monetare, în timpul testării 15 unități monetare, iar dacă defectul este descoperit după livrarea produsului, corectarea acestuia poate să ajungă să coste între 60 și 100 unități monetare (Figura 1)[3]. Prin urmare, cu cât se începe mai repede testarea sistemului, cu atât cresc șansele ca eventualele erori să fie descoperite din timp, fără să aibă un impact negativ asupra costului.

Conceptul modelului V a fost dezvoltat simultan, dar independent în Germania și în Statele Unite ale Americii la sfârșitul anilor '80 [2]. Modelul german a fost dezvoltat de IABG în Ottobrunn, în colaborare cu Oficiul Federal de Tehnologie de Apărare pentru Ministerul Apărării. În vara anului 1992 proiectul a fost preluat de Ministrul Federal de Interne [1]. Modelul SUA a fost prima dată documentat în 1991 la Consiliul Național al Ingineriei Sistemelor, și a fost dezvoltat pentru sisteme de sateliți, care implicau sisteme hardware, software și interacțiune cu un utilizator uman [2].

7.1. Modele de dezvoltare software – V-Model

7.1.2. Prezentare generala

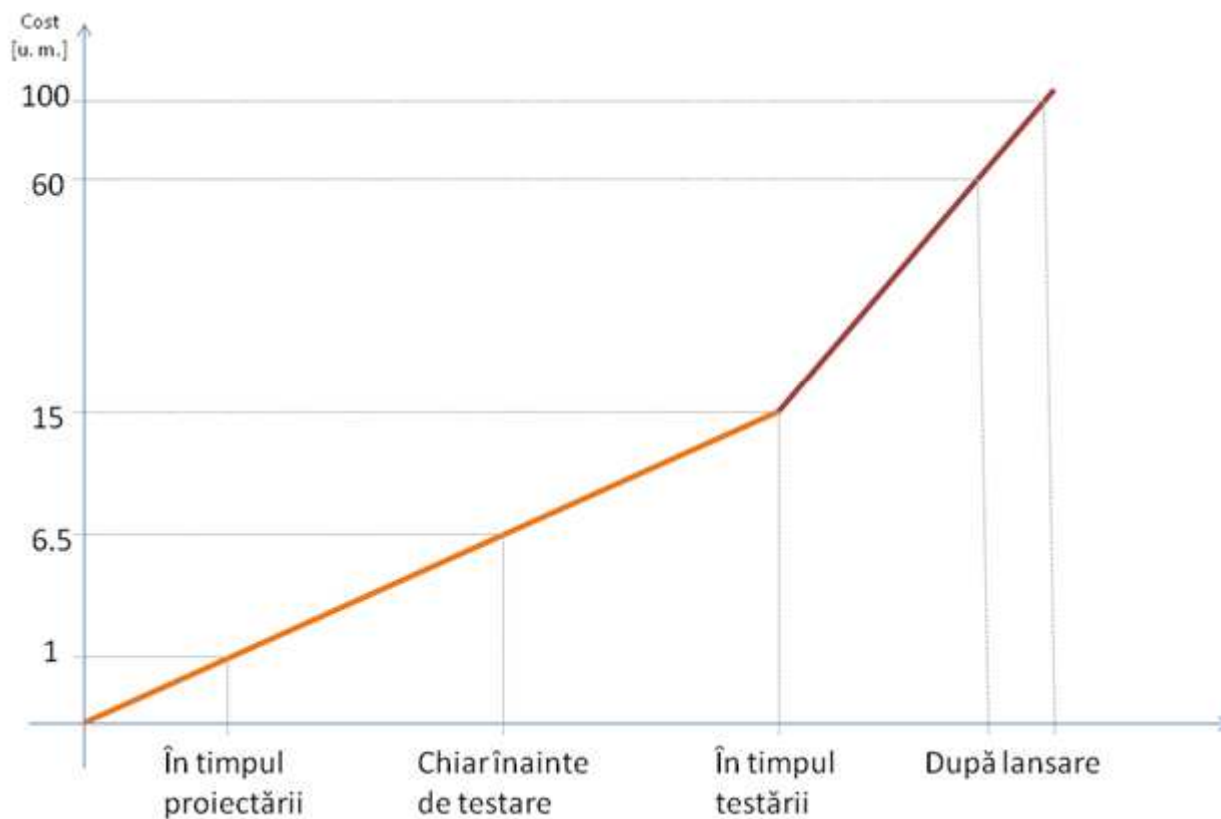


Figura 1 - Costul corectării defectului în funcție de când este descoperit

7.1. Modele de dezvoltare software – V-Model

7.1.3. Etapele modelului V

Preluarea specificațiilor

În faza de preluare a specificațiilor (Requirements Analysis), sunt colectate cerințele sistemului cerut prin analiza nevoilor utilizatorului. Acest pas nu determină însă cum o să fie proiectat sau construit software-ul. În acest pas, utilizatorul este intervievat, și este generat un document numit "Cerințele utilizatorului". Dacă este vorba de un proiect într-un domeniu în care inginerii ce vor construi sistemul nu au cunoștințe profunde, de regulă se va utiliza o echipă mixtă alcătuită din ingineri software cât și din specialiști din domeniul respectiv. Documentul creat în urma acestui proces va descrie cerințele utilizatorului în ceea ce privește funcționarea sistemului, interfețele cu utilizatorii, performanța acestuia, datele de intrare și ieșire, măsuri de siguranță necesare, costurile în care trebuie să se încadreze sistemul precum și orice alte cerințe. Utilizatorul are obligația de a verifica acest document, el stând după aceea ca și bază în proiectarea mai departe a sistemului. Tot pe baza acestui document, în această fază, testerii pot deja proiecta metodele de testare ce se vor implementa în *testul de acceptare de către utilizator* (user acceptance test).

Cerințele sistemului

Pe baza specificațiilor primite de la utilizator se vor formula cerințele pentru sistemul dezvoltat (System Specification). Dacă în prima etapă, cerințele sistemului era formulate, astfel încât să fie clar și pentru utilizator/client și pentru proiectanți, de această dată, specificațiile vor fi formulate stric într-un limbaj tehnic, folosind termeni care să fie bine înțeleși de către inginerii proiectanți, dar care s-ar fi putut să fie ambigue pentru utilizator. Pe baza acestor specificații se formulează testele ce se vor rula în faza de *testare a sistemului* (system testing).

7.1. Modele de dezvoltare software – V-Model

7.1.3. Etapele modelului V

Proiectarea sistemului

Urmând specificațiile impuse de pasul anterior, mai departe se trece la proiectarea sistemului (System Design). În acest pas se construiește *schema bloc* a sistemului ce urmează să fie realizat, urmând ca fiecare componentă a schemei bloc să fie analizată apoi de echipe specializate. Tot în acest pas se definesc modul prin care se vor interfața componentele între ele, cum ar fi de exemplu metode sau protocoale de comunicare folosite. Din acest pas vor rezulta testele necesare pentru a *testa interfațarea*, pe partea dreaptă a modelului V.

Proiectarea componentelor

În acest pas, sistemul fiind deja descompus în elemente bloc funcționale, se vor proiecta aceste elemente (Component Design). Aici, echipe specializate pun la punct fiecare detaliu ce ține de proiectarea componentelor electronice. Se stabilesc exact funcțiile ce le va executa componenta, se construiește schema principală a acestuia, se aleg componentele ce se vor folosi la realizarea lui și se face proiectarea PCB-ului. Din metodele de proiectare a componentelor precum și din specificațiile ce definesc funcțiile acestuia se vor deduce metode pentru *testarea componentelor*.

Realizarea componentelor

Pasul de realizare a componentelor (Component Construct) este cel mai de jos pas în modelul V. Acesta este ultimul pas care se execută, înainte de a se trece la testarea sistemului. În această etapă se execută plăcutele cu cablaj imprimat și acestea se echează conform proiectării ce s-a realizat în pasul anterior. După acest pas se începe "urcarea virtuală" pe partea dreaptă a modelului V, și începe practic procedura de testare.

7.1. Modele de dezvoltare software – V-Modell

7.1.3. Etapele modelului V

Desigur, gradul în care un sistem trebuie testat, și procedurile care se folosesc pentru testarea acestuia variază la fiecare proiect și depind de caracteristicile sistemului, precum și de modul în care sistemul a fost construit. Totuși, un principiu al testării poate fi urmat pentru toate sistemele și pașii ce trebuie urmați sunt prezentați în continuare, pe modelul V.

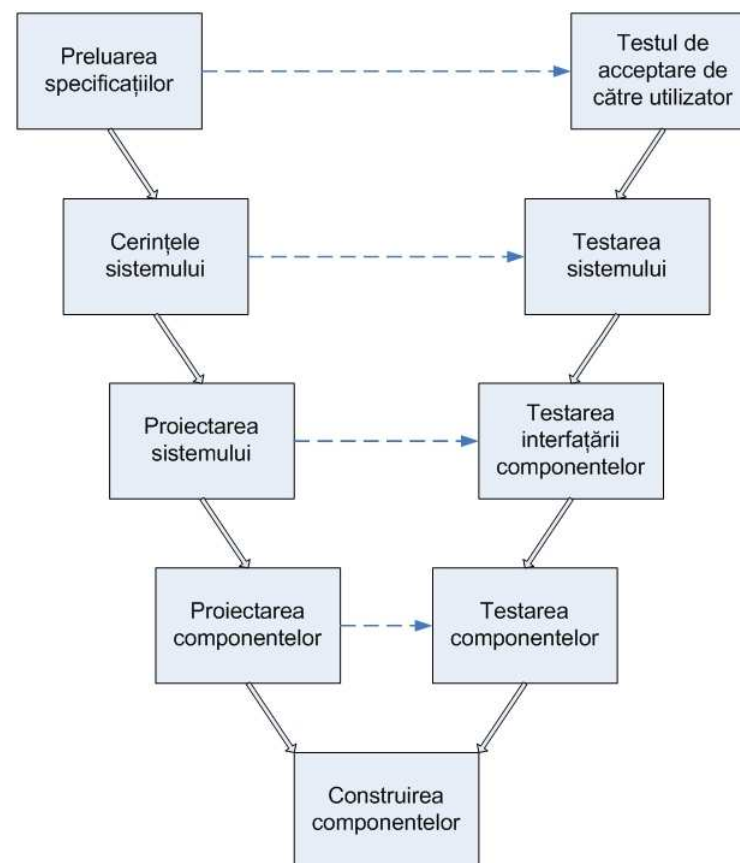


Figura 2 - Etapele modelului V

7.1. Modele de dezvoltare software – V-Model

7.1.3. Etapele modelului V

Testarea componentelor

Testarea componentelor (Component Testing) este practic primul pas de testare ce se realizează. În această fază testerii execută testare de tipul "white-box" și ei trebuie să se asigure că componenta îndeplinește toate cerințele de funcționare în condiții normale cât și în condiții de funcționare la limită. Teoretic, toate procesele de testare, deci și testarea componentelor, ar trebui să fie realizate de testeri independenți. În practică însă, de cele mai multe ori, din lipsă de personal sau de fonduri, de cele mai multe ori testarea componentelor se face de echipa care l-a dezvoltat, în timp ce se finalizează fazele de proiectare, pentru a se asigura ca componenta funcționează în mod corespunzător [4]. Acesta nu este metoda optimă de testare, însă în cele mai multe cazuri este realitatea.

Testarea interfațării componentelor

După ce componentele au fost construite, și în pasul anterior s-a asigurat că independent funcționează corect, următorul pas este de a le conecta împreună, pentru a se asigura că se interfațează conform standardelor prescrise (Interface Testing). Acest test nu este focalizat pe ce fac exact componentele ci mai degrabă cum comunică acestea împreună. Acesta verifică cum tratează componentele informațiile primite de la celelalte componente în condiții de funcționare normală, precum și cum răspund acestea la condiții de funcționare anormale, cum ar fi mesaje de eroare. Testul trebuie să fie organizat astfel încât fluxul de date să parcurgă întregul sistem, de la prima componentă până la ultima, precum trebuie testată și fiecare interfață de comunicație independent de celelalte.

7.1. Modele de dezvoltare software – V-Model

7.1.3. Etapele modelului V

Testarea sistemului

În acest pas este prima dată când întregul sistem este disponibil pentru testare ca și o entitate întreagă. Ca sistemul să fie admis, acesta trebuie să îndeplinească cerințele stabilite de specificațiile de sistem, și în această fază testarea se execută obligatoriu de o echipa de testeri independenți. Testarea trebuie să se facă atât pentru condiții de funcționare normală, cât și pentru condiții de funcționare sub stres, la limită și pentru a testa sistemele de protecție impuse, în condiții de utilizare anormală. De asemenea sistemul se va testa și peste pragul de toleranță precizat, pentru a vedea în ce condiții cedează și a stabili timpul necesar reparări sau înlocuirii acestuia, precum și eventualele posibilități de recuperare a datelor (unde este cazul). Tot în acest pas se verifică și documentația furnizată la sistem. Se verifică dacă acesta este suficient de clară și dacă tot ceea ce este descris în acesta se aplică întocmai sistemului.

7.1. Modele de dezvoltare software – V-Modell

7.1.3. Etapele modelului V

Testul de acceptare de către utilizator

Acest test ((User) Acceptance Testing – UAT), verifică sistemul în funcție de cerințele utilizatorului, dat fiind faptul ca până în acest punct toate verificările se făceau în funcție de cerințele stabilite de către echipa de ingineri la demararea proiectării. Este similar cu testarea sistemului, diferența fiind schimbarea focusului de la proiectant la utilizator. Testarea sistemului verifică dacă sistemul cerut a fost livrat. Testul de acceptare verifică dacă sistemul livrat a fost cel care a fost cerut de utilizator.

În acest punct, cei mai buni testeri sunt însăși utilizatorii care vor lucra cu noul sistem și care cunosc deja cum funcționează vechiul sistem dacă este vorba de un update, sau știu exact ce este așteptat ce la acesta dacă vorbim de un sistem complet nou. Prin urmare, pentru a face testarea, un eșantion de utilizatori este ales pentru a face testarea produsului în această fază. Această formă de testare se mai cheamă și *”testare beta”* (Beta testing) sau *”testare pe utilizatorul final”* (end user testing).

Conceptul pe care se bazează acest tip de testare este folosirea sistemului în modul în care și utilizatorul ar face-o, incluzând fiecare proces și funcționalitate a acestuia. Aici se testează folosirea sistemului în situații cu care numai utilizatorii familiarizați cu sistemul sunt capabili să le reproducă, situații pe care inginerii proiectanți nu aveau cum să le anticipeze și să le testeze. De obicei, un sistem bine definit și bine dezvoltat va trece cu brio acest gen de test.

7.1. Modele de dezvoltare software – V-Model

7.1.4. Obiectivele modelului V

Modelul V oferă ghidaj pentru planificarea și execuția proiectului. Următoarele scopuri sunt urmate să fie atinse prin utilizarea acestui model în timpul execuției:

- Minimizarea riscurilor asupra proiectului: modelul V îmbunătățește transparența proiectului și controlul acestuia. Prin urmare permite o recunoaștere timpurie a deviațiilor de la planul de proiectare, și ușurează managementul procesului
- Îmbunătățirea și garanția calității: ca și un model de proces standardizat, modelul V garantează că produsul livrat este complet și are calitatea cerută.
- Reducerea costului total pe durata întregului proiect: efortul pentru dezvoltare, producție, operare și mentenanță pentru sistem pot fi calculate și controlate într-o manieră transparentă.

7.1. Modele de dezvoltare software – V-Model

7.1.5. Neajunsurile modelului V

Desigur, ca orice model, și modelul V are anumite limitări și neajunsuri. Într-un articol publicat de firma "Claro Testing", intitulat "The Dangerous and Seductive V Model" [5], sunt prezentate majoritatea acestora.

Una dintre aceste neajunsuri este implicarea testerilor și mai ales a utilizatorilor prea târziu în proiect. Testul de acceptare de către utilizatori este executat abia pe produsul final, și prin urmare, orice neajuns este greu sau în cele mai multe cazuri chiar imposibil de corectat, fără reluarea proiectului aproape în întregime.

Tot implicarea târzie a testerilor duce și la reducerea semnificativă a timpilor acordați testării. Dacă un proiect este în întârziere, pentru a se livra produsul la termenul limită, timpii de testare vor fi scurtați față de planul original. Acesta va duce la o acoperire mai mică a testelor efectuate și prin urmare la livrarea unui sistem de calitate mai slabă.

Pentru a evita totuși aceste probleme, se propune încă de la început iterarea modelului V în mai multe etape. Astfel, se propune mai întâi construirea unui prototip al produsului care să fie testat până la testul de acceptare de utilizator, după care în cazul unor neajunsuri să fie reluat întregul model V, cu reproiectarea sistemului. Desigur, acest lucru necesită mult mai mult timp și mai mulți bani, și prin urmare, majoritatea firmelor clasează problemele raportate de utilizatorii finali ca fiind "de tip cosmetic" și nu se ocupă niciodată pe rezolvarea acestora [5].

7.1. Modele de dezvoltare software – V-Model

7.1.6. Modelul V multiplu

Modelul V multiplu vine întocmai în sprijinul ideii care susține că modelul V clasic trebuie iterat pentru fiecare stare de dezvoltarea a dispozitivului construit. Acesta se bazează pe ideea că pe durata procesului de dezvoltare diferitele stări fizice ale dispozitivului sunt produse: prima dată un model ce simulează comportamentul sistemului, apoi diferite prototipuri care evoluează apoi în produsul final. Toate aceste forme ale produsului respectă modelul V *clasic*, de unde vine denumirea de model V multiplu.

Sistemul este de obicei proiectat dintr-o secvență de prototipuri, care se apropie din ce în ce mai mult de produsul final. Mai întâi este construit un model al sistemului pe un calculator, care simulează comportamentul acestuia. După ce modelul este corect din punct de vedere funcțional, se generează cod din model, care apoi este implementat în prototip. Hardware-ul experimental al prototipului este apoi pas cu pas înlocuit cu hardware real, până când sistemul ajunge la forma sa finală, după care începe producția acestuia în masă. Motivul pentru care se folosesc acești pași intermediari în proiectare sunt faptul că a schimba prototipul este mult mai ieftin și durează mai puțin în timp decât să schimbi produsul final, și și mai ieftin e să schimbi ceva la model.

7.1. Modele de dezvoltare software – V-Model

7.1.6. Modelul V multiplu

Modelul V multiplu (Figura 3), bazat pe bine cunoscutul model V, este un model de dezvoltare care ia în considerare acest fapt. În principiu fiecare instanță a produsului (modelul, prototipurile și varianta finală) urmăresc un proces complet de dezvoltare a modelului V clasic, incluzând activitățile de dezvoltare, construire și testare. Acesta înseamnă de exemplu că funcționalitatea completă poate să fie testată atât pe modelul produsului cât și pe prototipurile sau varianta finală a acestuia. Pe de altă parte, unele proprietăți tehnice detaliate, cum ar fi impactul asupra mediului înconjurător, nu pot fi testate pe model foarte bine, și acestea trebuie să fie testate pe prototip.

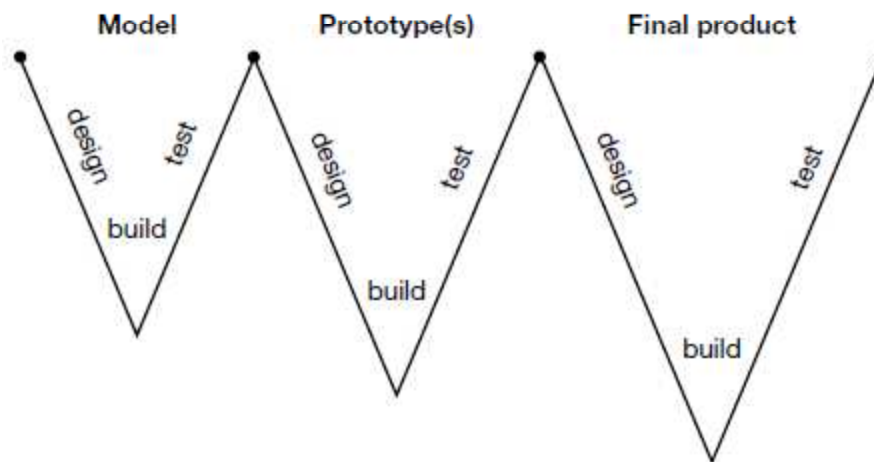


Figura 3 - Modelul V multiplu [6]

7.1. Modele de dezvoltare software – V-Modell

7.1.6. Modelul V multiplu

Procesul de testare implică o cantitate foarte mare de activități. Sunt multe tehnici de proiectare care vor fi aplicate, nivele și tipuri de test care trebuie executate, și în general, considerente legate de testare, care trebuie luate în considerare. Modelul V multiplu asistă în structurarea acestor activități. Prin alocarea lor pe ciclul V, oferă răspuns la întrebările: "Când pot activitățile să fie executate?", "Care parte a testării este cea mai relevantă în care stadiu al dezvoltării produsului?".

Organizarea testării în situația complexă a modelului V este însuși o sarcină dificilă. Deși într-adevăr detaliile sunt diferite la fiecare proiect în parte, se pot aplica însă aceleași principii pentru a structura activitățile de testare.

Mai departe sunt prezentate aranjarea diferitelor activități de testare în funcție de stadiul de dezvoltare în care se afla produsul.

7.1. Modele de dezvoltare software – V-Modell

7.1.6. Modelul V multiplu

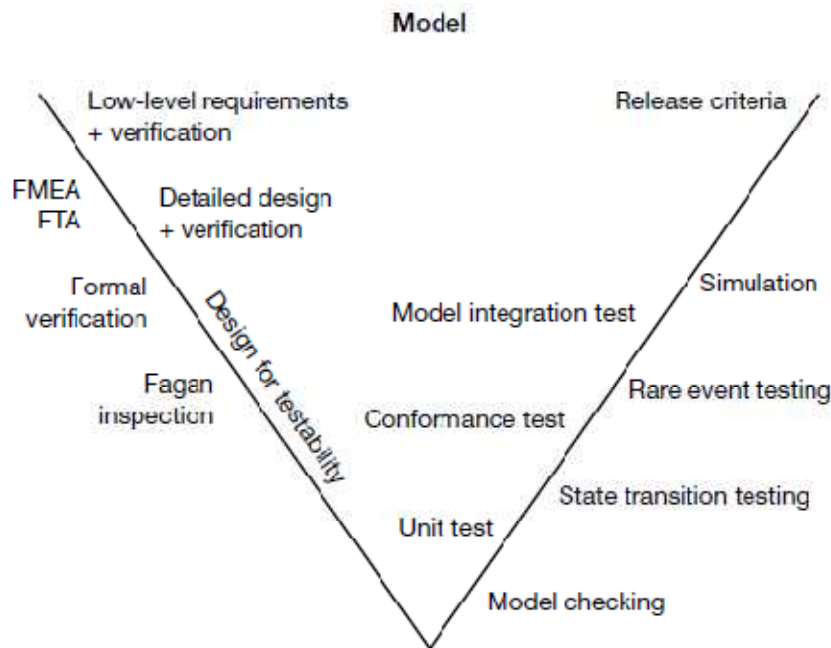


Figura 4 - Modelul V pentru stadiul de model al produsului [6]

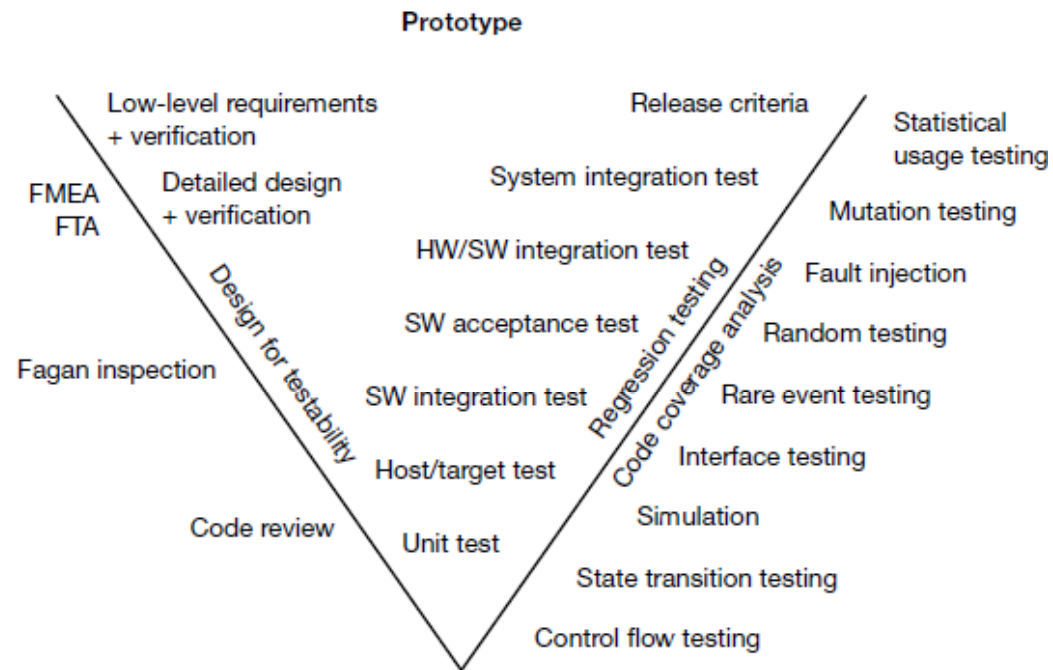


Figura 5 - Modelul V pentru stadiul de prototip al produsului [6]

7.1. Modele de dezvoltare software – V-Modell

7.1.6. Modelul V multiplu

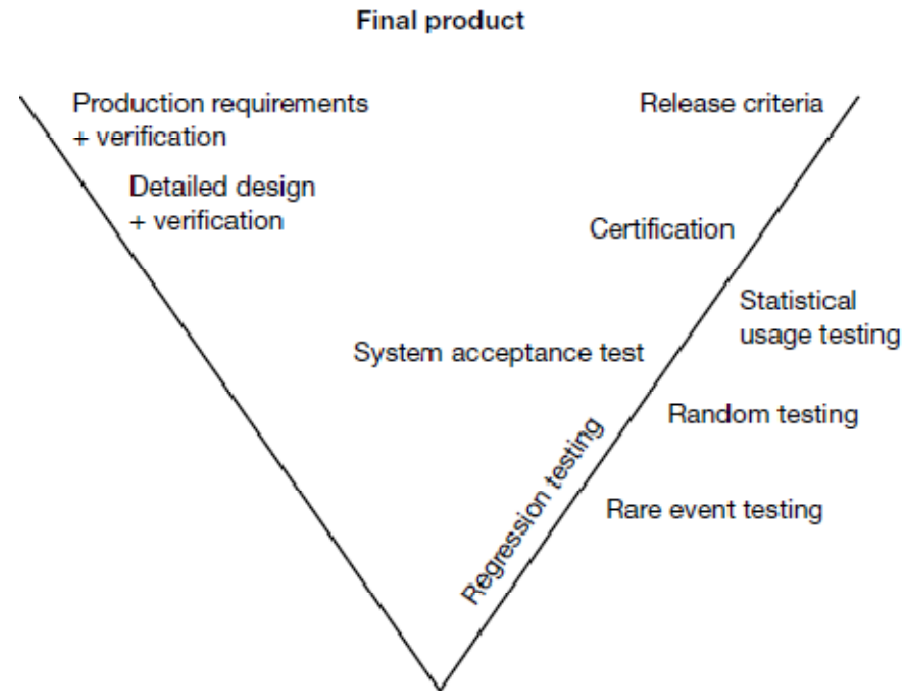


Figura 6 - Modelul V pentru stadiul final al produsului [6]

7.1. Modele de dezvoltare software – V-Modell

7.1.6. Modelul V multiplu

Modelul V multiplu în cei trei ciclii secvențiali V nu ia în considerare obiceiul de a descompune un sistem complex în unități funcționale mai simple. Dezvoltarea unui asemenea sistem începe prin stabilirea specificațiilor de proiectare la nivel înalt, urmat de o fază de planificare la nivelul arhitecturii, unde se determină ce componente (hardware și software) sunt necesare pentru realizarea proiectului. Aceste componente sunt mai departe dezvoltate separat, în final fiind integrate într-un sistem complet. Modelul V simplu poate fi aplicat acestui proces de dezvoltare la nivel înalt. Partea stângă a modelului se ocupă de descompunerea sistemului în componentele sale. Partea din mijloc a modelului V constă în ciclii de dezvoltare paralele pentru toate componentele, iar partea din dreapta a modelului V se ocupă de integrarea componentelor în sistem (Figura 7).

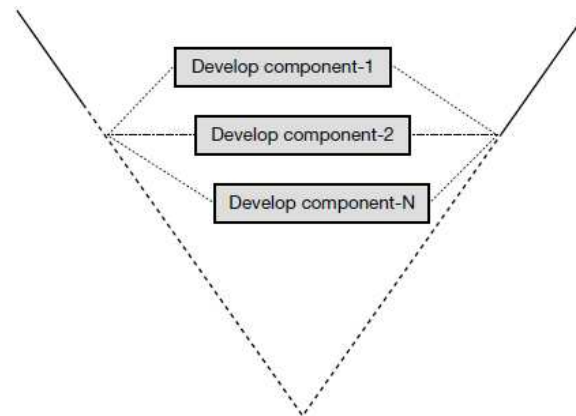


Figura 7 - Model V multiplu cu dezvoltarea paralelă a componentelor [6]

7.1. Modele de dezvoltare software – V-Modell

7.1.6. Modelul V multiplu

De fapt, modelul V pur secvențial este aplicabil numai la nivelul componentelor. De obicei nu sistemul complet se modelează întâi, apoi construit prototipul lui ș.a.m.d. Componentele sunt cele care sunt construite în pași așa. Acest fapt explică de ce unele activități de dezvoltare și unele probleme ce ar putea apărea nu pot fi bine plasate pe cele 3 V-uri ale modelului V, de exemplu cerințele la nivelul înalt și la nivelul inferior, planul de siguranță, și instrumentele specifice construirii. Acest lucru se explică prin faptul că acestea aparțin procesului global de dezvoltare.

Când modelul V la nivelul sistemului se combină cu modelul multiplu V la nivelul componentelor, rezulta așa numitul "model V încorporat" (Figura 8).

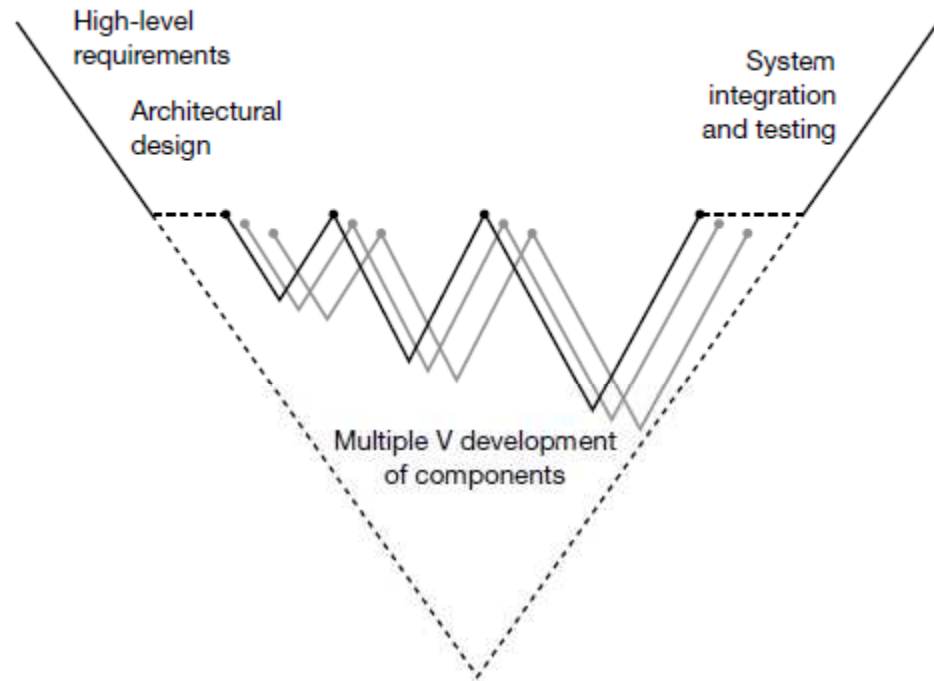


Figura 8 - Model V încorporat [6]

7.1. Modele de dezvoltare software – V-Modell

7.1.6. Modelul V multiplu

Cu acest model, toate activitățile legate de testare pot fi alocate nivelului corespunzător, la locul corect. Modelul V multiplu nu este util numai când trebuie proiectate și executate procedeele de testare pentru un produs complet nou, dar și în cazul lansării unei versiuni noi ai aceluiași produs.

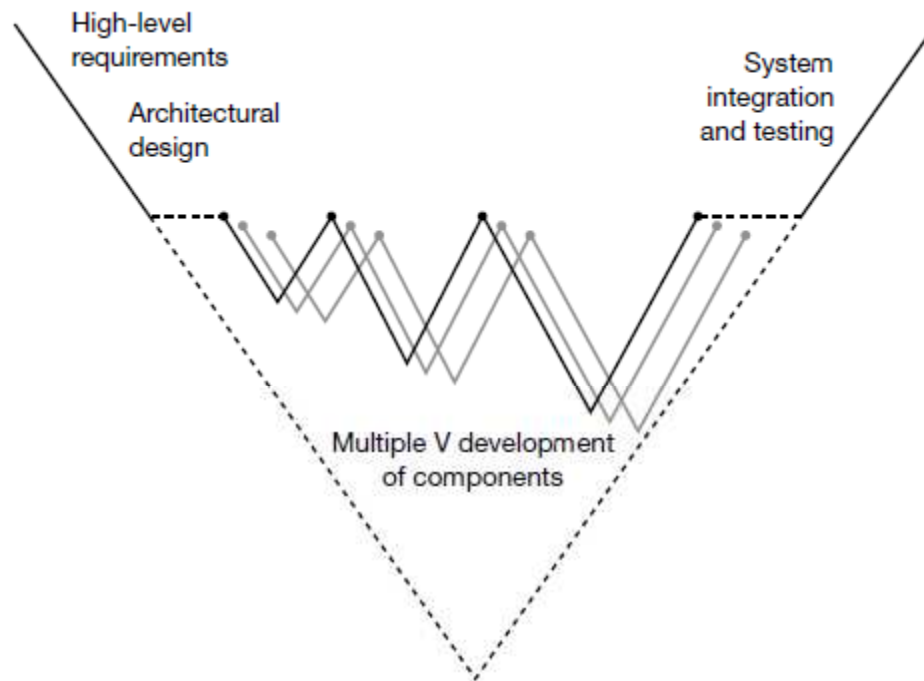


Figura 8 - Model V încorporat [6]