

# Inxhinieria Softuerike

## Konceptet baze të Inxhinieris Softuerike

---

Erzen Talla

# syllabusi – planifikim i ligjeratave dhe ushtrime

---

**Njësia I:** *Hyrje në Inxhinierinë e Softuerit dhe modelet e procesit për zhvillimin e softuerit (Metodologjia tradicionale dhe e shkathët SDLC)*

Hyrje në inxhinierinë e softuerit, fillimi i projektit, definimi i skopit projektit, analiza teknike dhe teknologjike (analiza e fizibilitetit), analizimi i procesit tradicional dhe të shkathët të softuerit.

**Njësia II:** *Analiza e Kërkesave dhe Përcaktimi dhe ri-inxhinierimi i modelit të kërkesave në Agile*

Përcaktimi dhe analizimi i kërkesës në qasjen Agile - Mbledhja e nevojave, Përcaktimi i kërkesave (duke përdorur tregimet e përdoruesve, aplikimi i 3C-ve dhe përcaktimi i kriterëve të pranimit), definimi i sprintave dhe zbatimi i aktiviteteve të tij, klasifikimi i kërkesave, Diagrami i rasteve të përdorimit dhe skenarëve.

**Njësia III:** *Aplikimi i UML për Dizajnimin dhe modelimin e Sistemit/Programit*

Përcaktimi i arkitekturës së sistemit, Përcaktimi i arkitekturës së të dhënave. Aplikimi i gjuhës së unifikuar të modelimit (UML) për dizajnimin e sistemit (përkufizimi i moduleve dhe entitetet e tij - modulari), definimi i diagramit të klasës UML dhe vetitë e tyre, Diagramet e aktiviteteve, GUI-ja e dizajnit.

**Njësia IV:** *Verifikimi, vërtetimi dhe vendosja e sistemit*

Testimi dhe evolucioni i softuerit, Zbatimi i vendosjes (deploy) së sistemit softuerik

# Planifikimi për Lëndën: *syllabusi – vlersimi*

---

## □ Aktiviteti i Vlerësimit

- Projekti= 40 pike
- Detyra/aktiviteti ne klase = 10 pike
- Testi Final = 50pike (Gjate nje viti akademik= 3 afate te rregullt te provimit)

## □ **Detyra** dhe **aktiviteti** ne ligjerata vlersohet vetem nese jeni present ne ligjerata

## □ Studenti **përsëritës** nuk obligohet të jete pjesmarrs ne ligjerata por obligohet te i perfundoj te gjitha aktivitet (pjeset e projektit) te parapar.

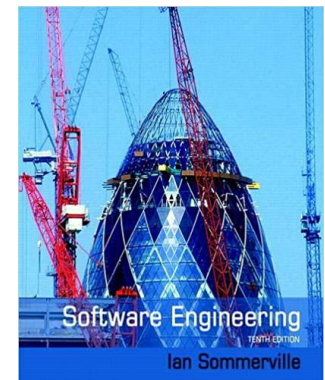
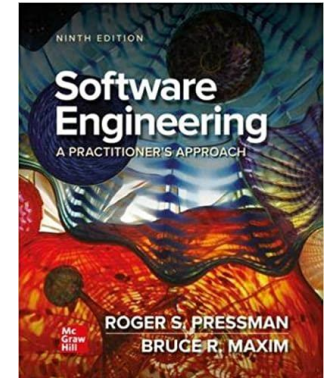
- Kualefikohen per (nota maximale 8-tete:
  - 50-65 pike nota 6 Gjashte
  - 66-85 pike nota 7 Shtate
  - 86-100 pike nota 8 Tete

# Referencat për Lënden

## □ Librat për Lëndën:

- Software Engineering: A Practitioner's Approach 9th Edition by Roger Pressman (Author), Bruce Maxim (Author)
- Software Engineering 9<sup>th</sup>/10<sup>th</sup> Edition, by Ian Sommerville , 2015.
- Systems Analysis and Design in a Changing World, Sixth Edition John W. Satzinger, Robert B. Jackson, Stephen D. Burd. 2012.

## □ Moodle (ligjeratat-PowerPoint, materiale-Ushtrime.. etj)



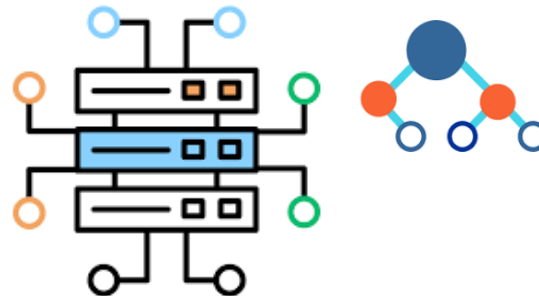
# Ç'ka është Softueri?

□ **Softueri** është :

- (1) **programet kompjuterike** (instruksione) që kur ekzekutohen ofrojnë **veçoritë** (features), **funksionin** dhe **performancën** e dëshiruar;
- (2) **strukturat e të dhënave** që mundësojnë që programet të **manipulojnë** në mënyrë **adekuate** informacionin;
- (3) **informacion përshkrues** si në formë të **fizike** ashtu edhe në formë **virtuale** që **përshkruan** funksionimin dhe përdorimin e programeve.



computer programs



data structures



Documents Soft & Hard

# karakteristikat e një Softueri të mirë

---

□ Çdo **softuer** duhet të plotësojë atributet e mëposhtme:

- **OPERATIVE** - kjo karakteristikë matë se sa mirë punon softueri në operacionet si në:
  - Buxhet, Efikasitet, Përdorshmëri, Besueshmëri, Korrektësi, Funksionalitet, Siguri.
- **TRANZICIONALE** - zhvendosja nga një platformë në tjetrën apo nderthuer teknologji te ndryshme:
  - Ndërveprimi (Interoperability), Ripërdorimi (Reusability), Transportueshmëri (Portability), Përshtatshmëri (Adaptability)
- **MIRËMBAJTJA** – aftësia e softueri të përshtatet në mjedisin që ndryshon shpejt:
  - Fleksibil, Modular, Shkallëzueshëm, Mirëmbahet

## në përfundim të Lëndes

---

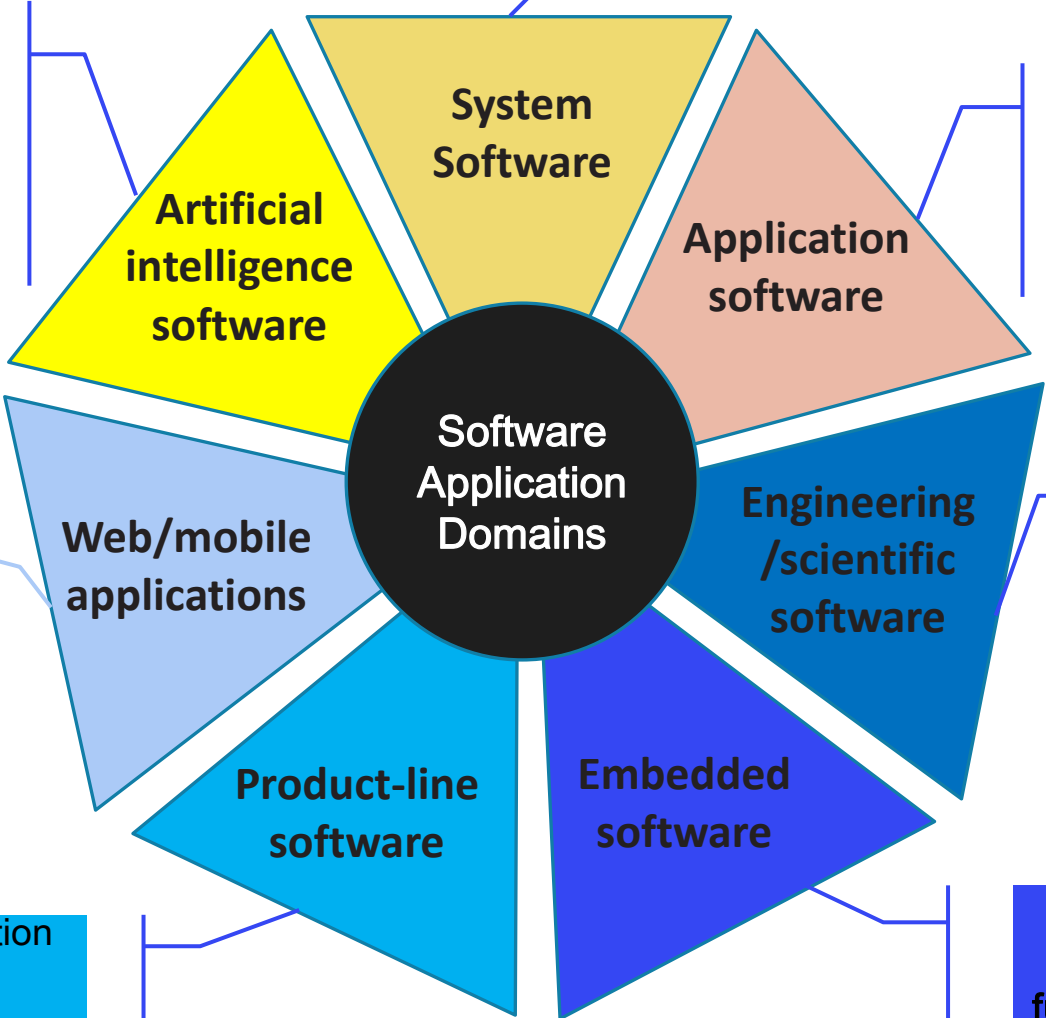
- ☐ **Aplikoni** dhe **zbatoni** *praktikat kryesore të inxhinierisë së softuerit* në nivelin konceptual për problemin e dhënë.
- ☐ **Krahasoni** dhe **vlersoni** *metodologjitë tradicionale, agile* dhe të dobëta të zhvillimit në një nivel të lartë.
- ☐ **Propozoni** një metodologji më të *përshtatshme* për një **situatë të caktuar**.
- ☐ **Implementoni** i metodologjin **Agile/scrum** në zhvillimin e softuerit
- ☐ **Zbatoni** *praktikat inxhinierike* për **krijimin** e softuereve *qualitative* dhe *cilsorë*.
- ☐ **Modeloni** **modulet, objektet** dhe **modelete e entiteteve** të një sistem softuerik
- ☐ *Njoftohuni* me procesin e **Verefikimit** dhe **Vlersimit** të **Qualiteti të Cilsise** së sistemit softuerike

# Domenet/fushat e Aplikacioneve Softuerike

Applications included in robotics, decision-making systems, pattern recognition (image and voice), machine learning.  
e.g., AI in Autonomous Vehicles, AI education, etc.

A wide array of applications browser-based apps, cloud computing, service-based computing, and software that resides on mobile devices.  
e.g., Skype, Facebook, Linked-in, online shopping etc.

A software product line is a collection of multiple software.  
e.g., inventory control products, accounting software, POS, etc.



A programs to serve other programs. e.g., Operating systems, device drivers, compilers, editors, networking software, BIOS, telecommunication processors

Stand-alone programs that solve a specific business need.  
e.g., Customized Software, Microsoft Office, Internet Browsers, Media Players, Antiviruses, Point of Sale, etc

Such software is written for specific applications using principles, techniques. e.g; MATLAB, AUTOCAD, PSPICE, ORCAD, etc.

e.g., key pad control for a microwave oven, digital functions in an automobile such as fuel control, dashboard displays, and braking system, etc.



# krizat e Softuerit

---

□ **Krizat e softuerit** (termi u përdor në fund të viteve 1960) - shumë kompani *nuk arritën të zhvillonin projekte softuerike*, ka pasur shumë arsye të krizave të softuerit përfshijnë:

- Projekti tejkalon buxhetin.
- Koha e projektit mbaron, gjë që shkakton dërgimin e vonuar të softuerit
- Cilësi e dobët e softuerit
- Softueri shpesh nuk i plotësonte kërkesat
- Projekti nuk menaxhoi mirë dhe ishte shumë joefikas
- Kompleksiteti i softuerit rritet do të thotë që softueri ishte kompleks për përdoruesit
- Kostoja e softuerit rritet në krahasim me harduerin
- Mungesa e komunikimit ndërmjet zhvilluesve të softuerit dhe përdoruesve.
- Mungesa e trajnimit adekuat në inxhinieri softuerike.

krizat e Softuerit...!

❑ **\$81 miliardë dollarë**

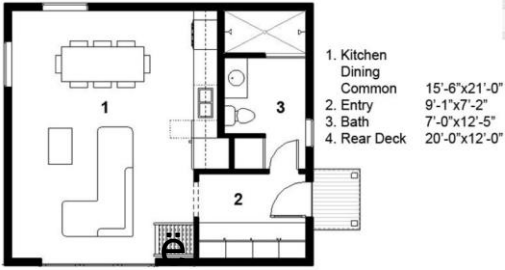


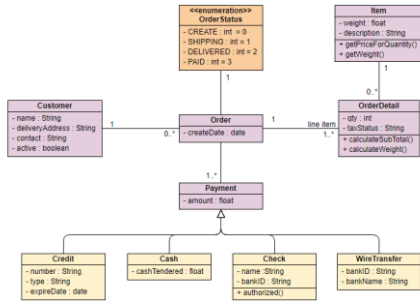


❑ **HUMBJET në PROJEKTET e ANULUARA**

❑ **Burimi: Raporti i Kaosit të Grupit STANDISH**

TRADITIONAL RESOLUTION FOR ALL PROJECTS					
	2011	2012	2013	2014	2015
SUCCESSFUL	39%	37%	41%	36%	36%
CHALLENGED	39%	46%	40%	47%	45%
FAILED	22%	17%	19%	17%	19%

*The Traditional resolution of all software projects from FY2011–2015 within the new CHAOS database.*

# Konceptet e Inxhinierisë Softuerike

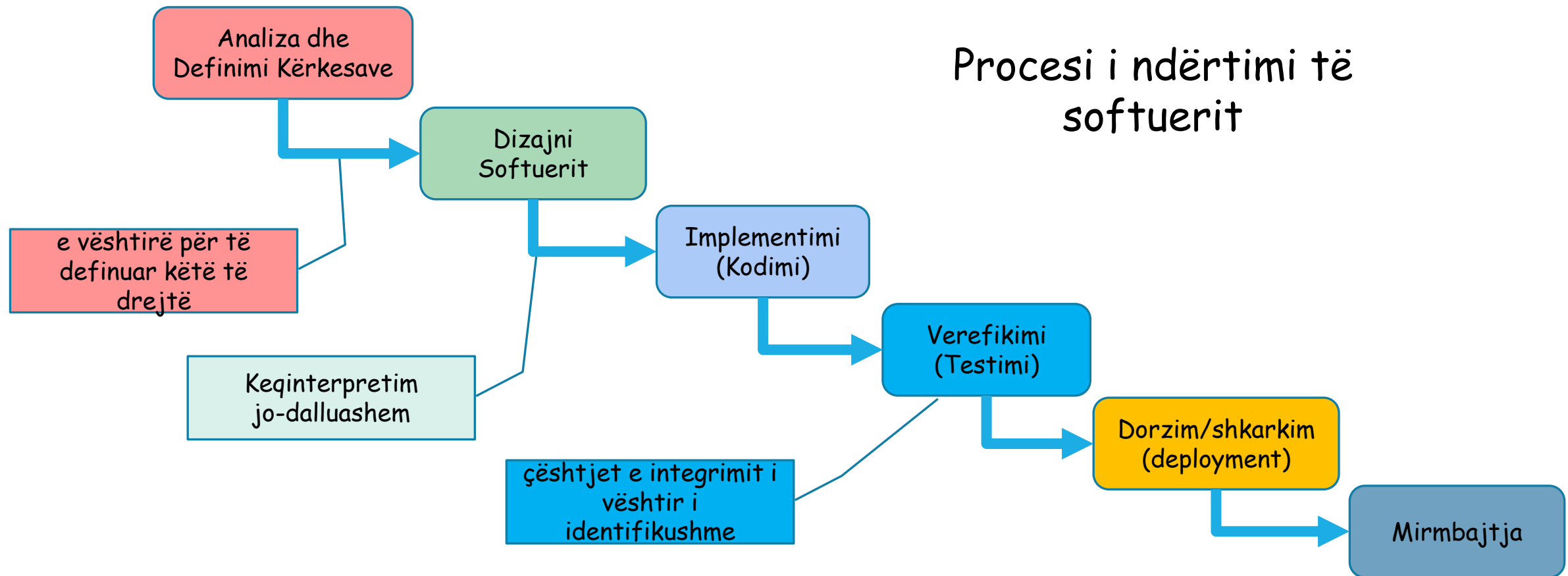
Engineering	 <p>1. Kitchen Dining Common 15'-6"x21'-0"          2. Entry 9'-1"x7'-2"          3. Bath 7'-0"x12'-5"          4. Rear Deck 20'-0"x12'-0"</p>		
	Dizajni	Ndertimit	Produkti
Software Engineering	 <pre> classDiagram     class Customer {         -name: String         -deliveryAddress: String         -contact: String         -active: boolean     }     class Order {         -createDate: date     }     class Item {         -weight: float         -description: String         +getPriceForQuantity()         +getWeight()     }     class OrderDetail {         -qty: int         -taxStatus: String         +calculateSubTotal()         +calculateWeight()     }     class Payment {         -amount: float     }     class Credit {         -number: String         -type: String         -expireDate: date     }     class Cash {         -cashTendered: float     }     class Check {         -name: String         -bankID: String         +authorize()     }     class WireTransfer {         -bankID: String         -bankName: String     }      Customer "1" -- "0..1" Order : -createDate: date     Order "1" -- "1..*" Payment     Order "1" -- "0..*" Item : line item     Order "1" -- "1..*" OrderDetail     Payment "1" -- "1..*" Credit     Payment "1" -- "1..*" Cash     Payment "1" -- "1..*" Check     Payment "1" -- "1..*" WireTransfer         </pre>		

# Konceptet e Inxhinierisë Softuerike...



# Konceptet e Inxhinierisë Softuerike...

## Procesi i ndërtimi të softuerit



# Konceptet e Inxhinierisë Softuerike...

---

- ❑ **Qëllimi i Inxhinierisë Softuerike** është të **prodhojë** zgjidhje softuerike të **qendryshme** (robust) dhe me cilësi të lartë:
  - në kohë,
  - në buxhet
  - që ofrojnë vlerë për përdoruesit.
  
- ❑ **Objektivi** i inxhinierisë softuerike është të **prodhojë** një softuer që do të jetë efikas, i besueshëm, i përdorshëm, i modifikueshëm, i transferueshëm (në platforma ndryshme), i testueshëm, i ripërdorshëm, i mirëmbajtur, i ndërveprueshëm dhe i saktë.

# Konceptet e Inxhinierisë Softuerike...

---

## □ Detyrat tipike të inxhinierisë softuerike

- Analiza e problemit, Përcaktimi i kërkesave, Dizajnimi i softuerit, Kodimi i zgjidhjes softuerike, Testimi dhe integrimi i kodit, Instalimi dhe vendosjen e softuerit, Dokumentacioni, Mirëmbajtja, sigurimi i cilësisë, Trajnimi, Vlerësimi i burimeve, Menaxhimi i projektit.

## □ Cikli i Jetësimit (Life Cycle) të Softuerit

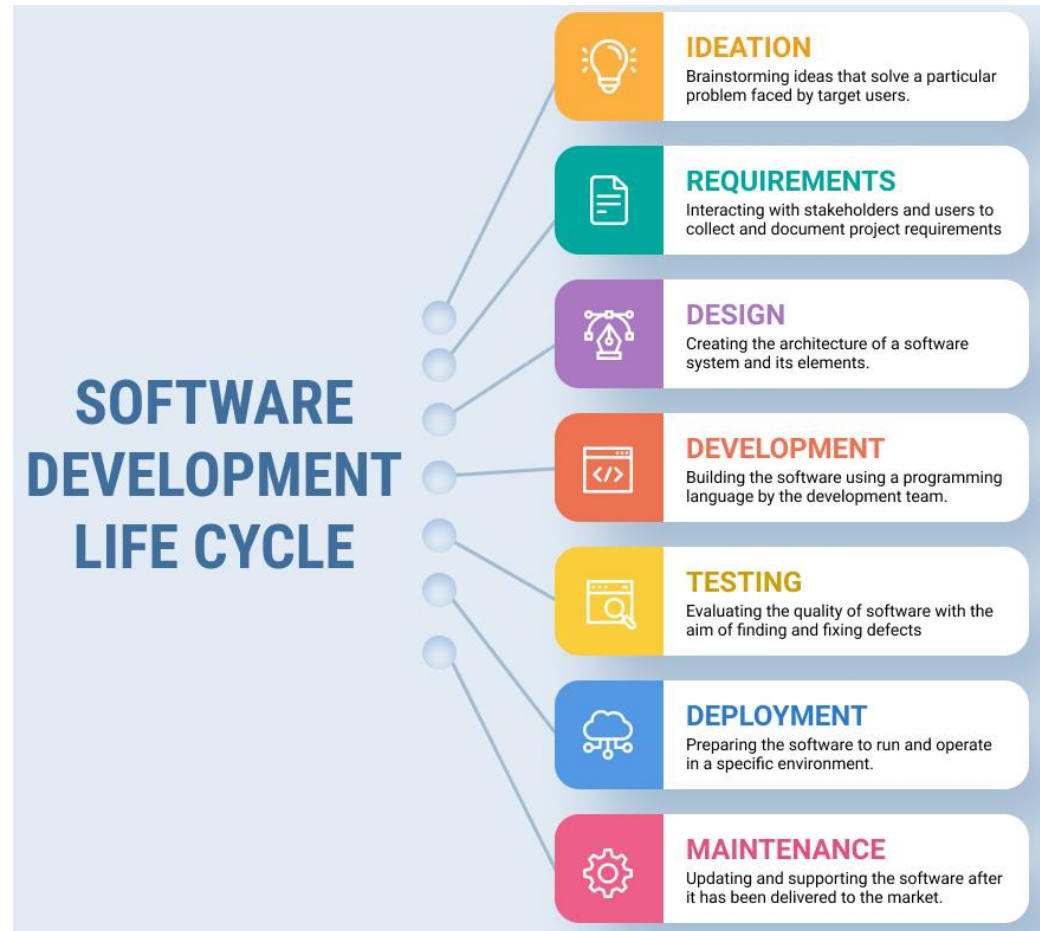
- Cikli i Jetësimit të zhvillimit të softuerit (SDLC) i referohet një metodologjie ose fraworke me procese të përcaktuara qartë për krijimin e zgjidhjeve të softuerit me cilësi të lartë në buxhet dhe në kohë.
- Zakonisht ndahet në gjashtë deri në shtatë hapa/aktivitete:
  - Planifikimi (inicimi), Kërkesat, Dizajnimi, Ndërtimi, Testimi, Vendosja (deploy), Mirëmbajtja

# Cikli i Jetësimit (Life Cycle) të Softuerit

## □ Fazat e SDLC

- **aktivitetet** e ndryshme që janë zakonisht pjesë e procesit *sistematik* të **inxhinierisë softuerike**.

<https://computools.com/software-development-life-cycle-sdlc/>





# Cikli i Jetësimit (Life Cycle) të Softuerit...

1

## Requirements Gathering

The team identifies, gathers, and defines current problems, requirements, and requests related to the software application or service.

2

## Software Design

The team makes software design decisions regarding the architecture and make of the software solution based on gathered requirements and criteria.

3

## Software Development

Teams build the software solutions based on the design decisions made.

4

## Test & Integration

The software is tested and checked for quality and performance acceptance.

5

## Deployment

The software is deployed into a production environment. This process includes provisioning and configuring infrastructure, and defining a software deployment strategy.

6

## Operationalization & Maintenance

The software is operationalized to monitor, control, and detect performance issues or incidents related to the deployment.

# Cikli i Jetësimit (Life Cycle) të Softuerit...

Software Development Life Cycle (SDLC)



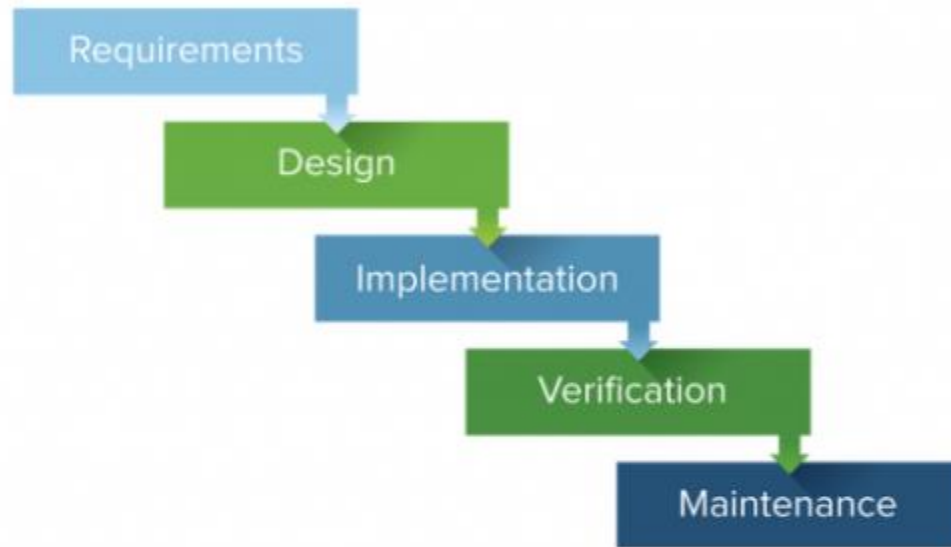
5 | SlideSalad.com

slidesalad

# Cikli i Jetësimit (Life Cycle) të Softuerit...

- ❑ Mund të ketë **mënyra** të ndryshme se si **kryhen** **aktivitetet** (SDLC) nga kjo varet:
  - **Skopi**/fushveprimi (scop) **Projektit**,
  - **Qasja** (approach) e **Organizates**,
  - **Ekipi i Zhvillimit**.
- ❑ **Koha** e këtyre aktiviteteve dhe zgjedhja midis **metodave përsëritëse** (iterative) dhe **jo përsëritëse** (noniterative) shpesh **përkruhen** nga ato që njihen si **modele të zhvillimit të softuerit**:
  - Modeli **Linear** (ujvar/waterfall)
  - Modeli **Iterative**
  - Modeli **Prototype**
  - Modeli **Spiral**
  - Modeli **RAD** (Rapid Application Development)
  - Modeli **Agile Model**
  - Modeli (i ri) **DevOps**

## Waterfall



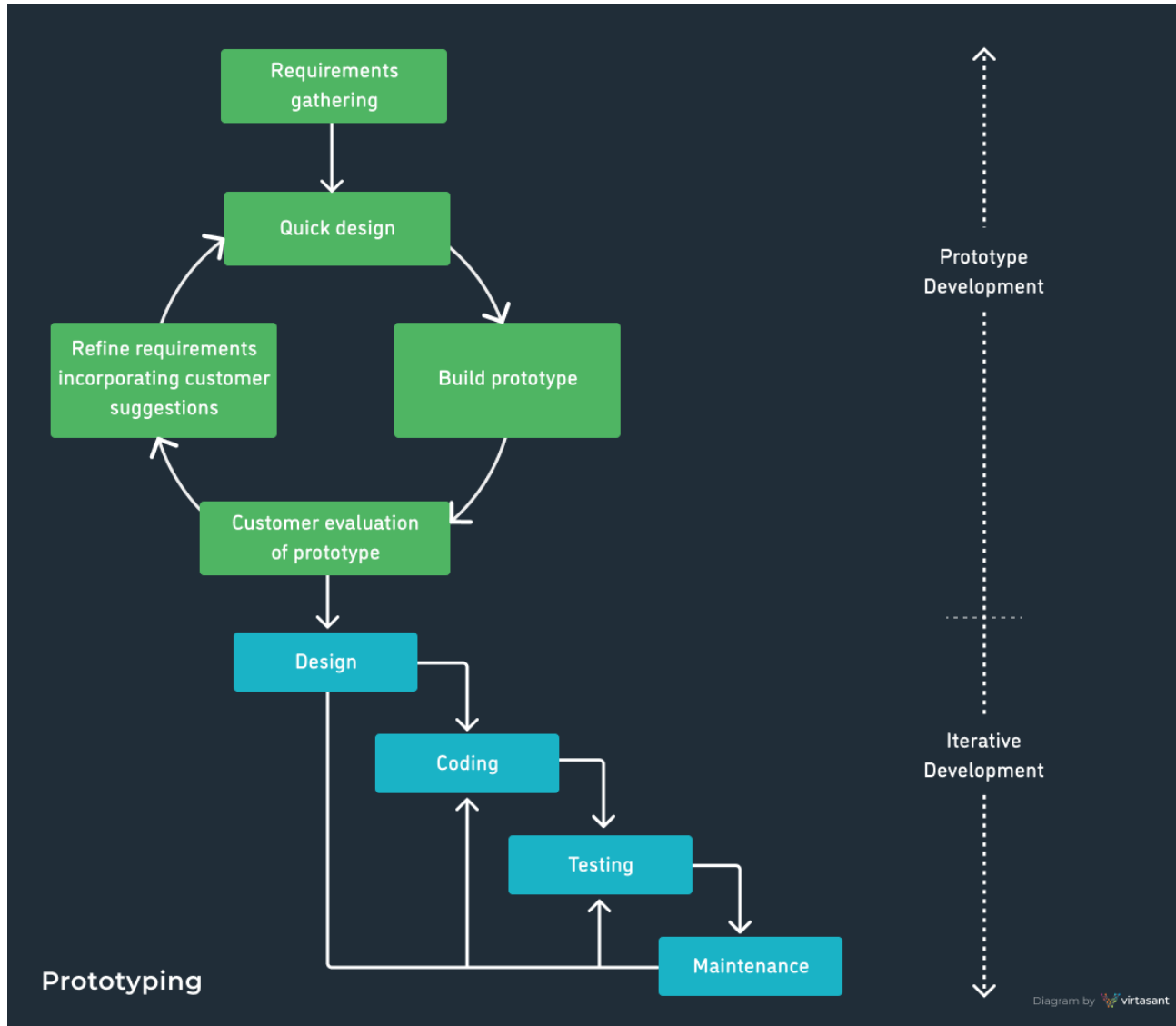
- Sequential/linear stages
- Upfront planning and in-depth documentation
- Contract negotiation
- Best for simple, unchanging projects
- Close project manager involvement

## Agile

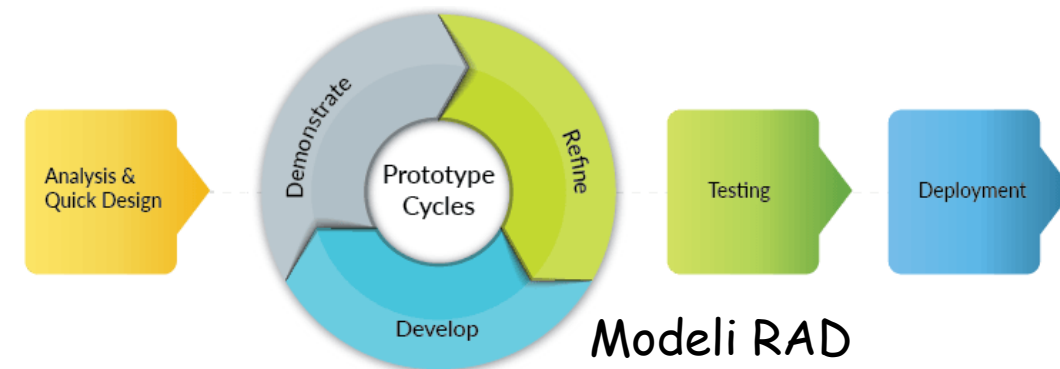
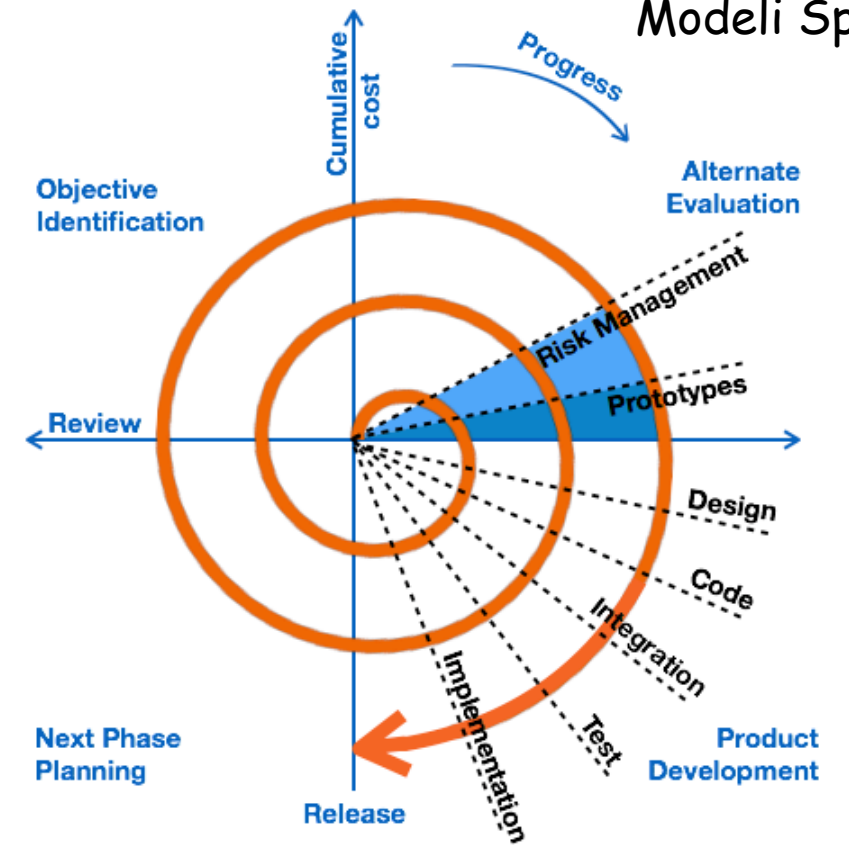


- Continuous cycles
- Small, high-functioning, collaborative teams
- Multiple methodologies
- Flexible/continuous evolution
- Customer involvement

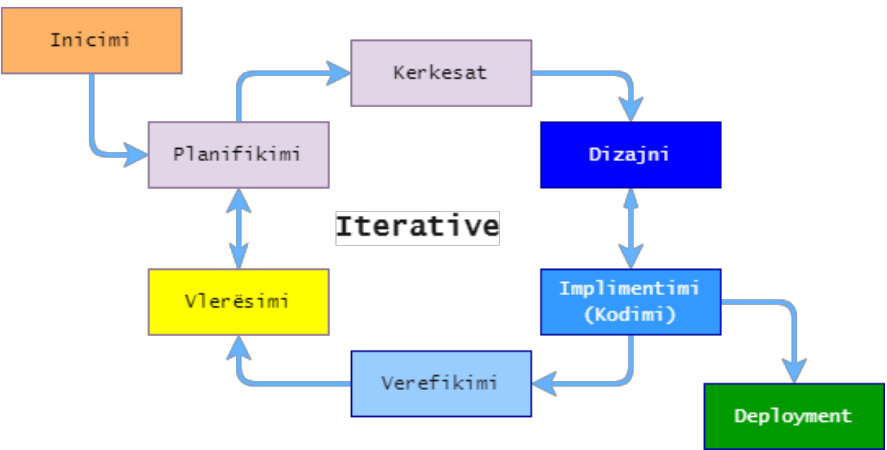
# Modeli Prototipit



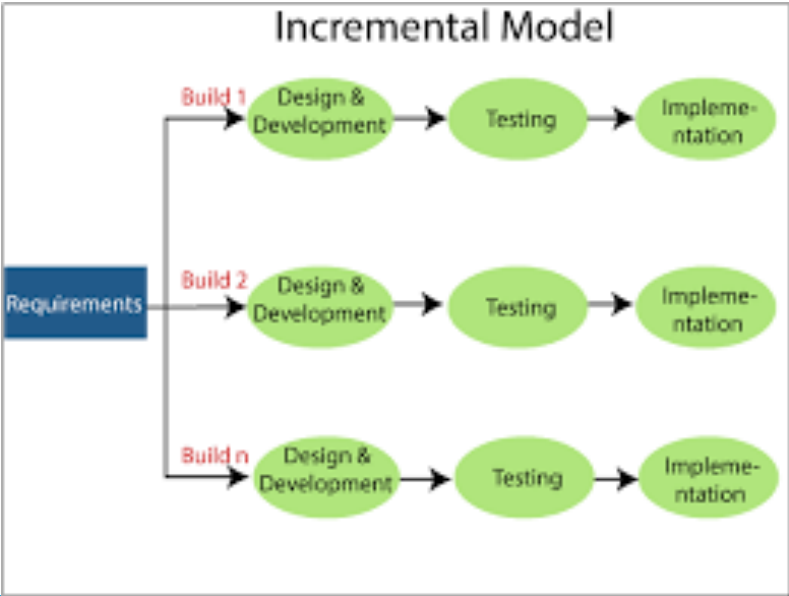
# Modeli Spiral



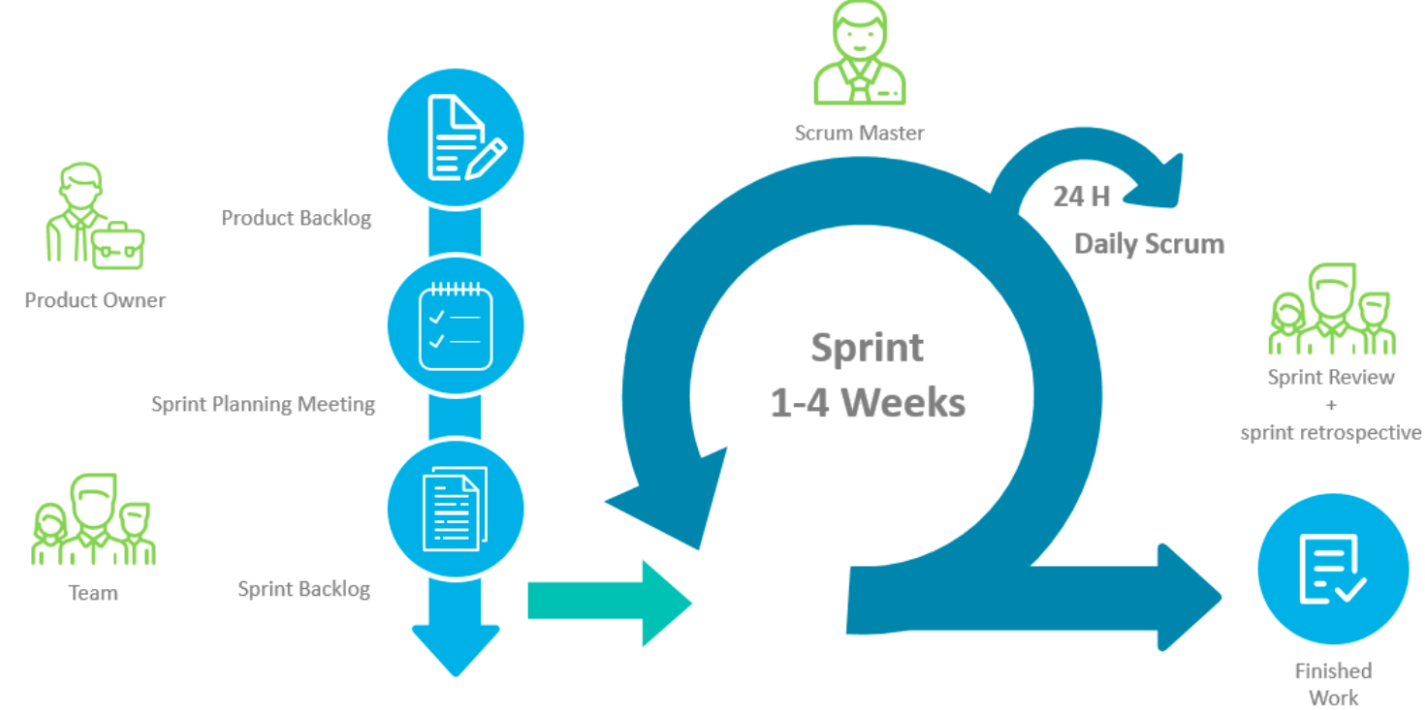
# Modeli RAD



Modeli Iterative (Perserites)



# AGILE SCRUM PROCESS





# kodimi 'NUK ËSHTE' Inxhinieri Softuerike

□ kodimi është pjesë

"prodhuese" e zhvillimit të softuerit.

□ Paramendoni se *çfarë duhet* për të ndërtuar një aeroplan.

- njerëzit që u kërkohet të ndërtojnë një aeroplan nga punëtorët e llamarinës, salduesit, njerëzit që *instalojnë* sistemet **hidraulike** dhe **elektrike** dhe të gjithë **sensorët**, etj.

- titullin "**inxhinier aeronautik**".

- Ky **titull** është i rezervuar për ata individë që projektuan aeroplanin dhe krijuan planet dhe ideuan proceset që teknikët dhe montuesit i **përdorin** për të ndërtuar aeroplanin.



Juice Flair/Shutterstock.com

# Inxhinieri Softuerike: Përkufizim

---

□ **Zhvillimi i aplikacioneve (programeve)** më të mëdha dhe komplekse kërkon qasje të mirëmenduara dhe një organizim të mirë.

- Si të organizohet,

dizenjimi dhe

implementimi i sistemeve kompjuterike të mëdha dhe komplekse

janë gjëra për të cilat *ju nevojitet* një qasje e inxhinierisë softuerike.



# Inxhinieri Softuerike: Përkufizim...

□ **Inxhinieria e Softuerit** është një koleksion **teknikash, metodologjish** dhe **mjetesh** që ndihmojnë në implementimin:

- e një *sistem softuerik* me **cilësi të lartë**
- me një **buxhet** të caktuar
- **para** ose **në** një **afati të caktuar** (deadline)
- **ndonëse** *ndodh* **ndryshimi**.

(1) IEEE [IEE17] ka zhvilluar përkufizimin e mëposhtëm për inxhinierinë e softuerit:

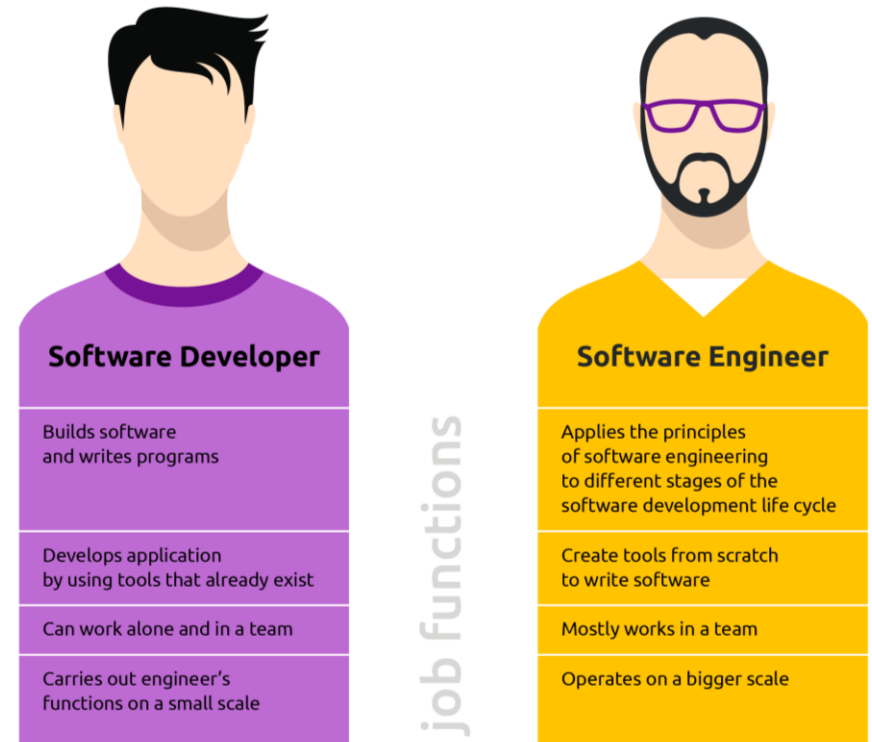
**Inxhinieri Softuerike**: Aplikimi i një **qasjeje sistematike**, të **disiplinuar**, të **matshme** për **zhvillimin, funksionimin** dhe **mirëmbajtjen** e softuerit; pra ky është aplikimi i **inxhinierisë** në **Softuer**.

(2) **Inxhinieria e softuerit** përfshin **funksionet, aktivitetet** dhe **detyrat**, duke *përfshirë* proceset e **zhvillimit, menaxhimit** e projektit, **analizën** e biznesit, **modelimin e kërkesave, dizajnet** e **përdorshmërisë, performancën operacionale, sigurinë, menaxhimit financiar, menaxhimit rregullator** dhe të **pajtueshmërisë, menaxhimit e rrezikut, sigurimit e cilësisë, kontrollin e cilësisë, menaxhimit e lëshimit (deployment)**, dhe **menaxhimit e shërbimit**

# Inxhinier i Softuerit vs. Zhvillues i softuerit

□ një **Zhvillues i Softuerit** di se si të **kodojë** dhe mund të ketë **aftësitë teknike** të nevojshme për të **ndërtuar** produkte kuptimplote. Ata duhet të kenë njohuri në **gjuhët specifike** kompjuterike të cilat mund t'i **ndihmojnë** ata të punojnë në mënyrë **efektive** me **programet kompjuterike**. "një Zhvillues ka **tendencë të punojë vetëm**".

□ një **Inxhinier Softuerësh** ndjek një **proces sistematik** të të kuptuarit të **kërkesave**, duke punuar me **palët e interesit** dhe duke **zhvilluar** një **zgjidhje** që plotëson **nevojat e tyre të klientëve** ose **përdoruesve**. Po ashtu **Inxhinier Softuerësh** ndihmojnë në zhvillimin e softuerit duke ofruar **zgjidhje** më të **përshtatshme teknologjike**. "një inxhinier softuerësh është pjesë e një ekipi më të madh"



**Faleminderit...!**

