

به نام پروردگار دانایی

# طراحی سیستم‌های شی گرا چرخه حیات توسعه نرم افزار

درس اول

---

Software Development  
Lifecycle (SDLC)

سید کاوه احمدی

## چرخه حیات توسعه نرم افزار (Software Development Lifecycle)

- روشی است برای مدیریت فعالیتهای مورد نیاز به منظور توسعه نرم افزارهای با کیفیت بالا.

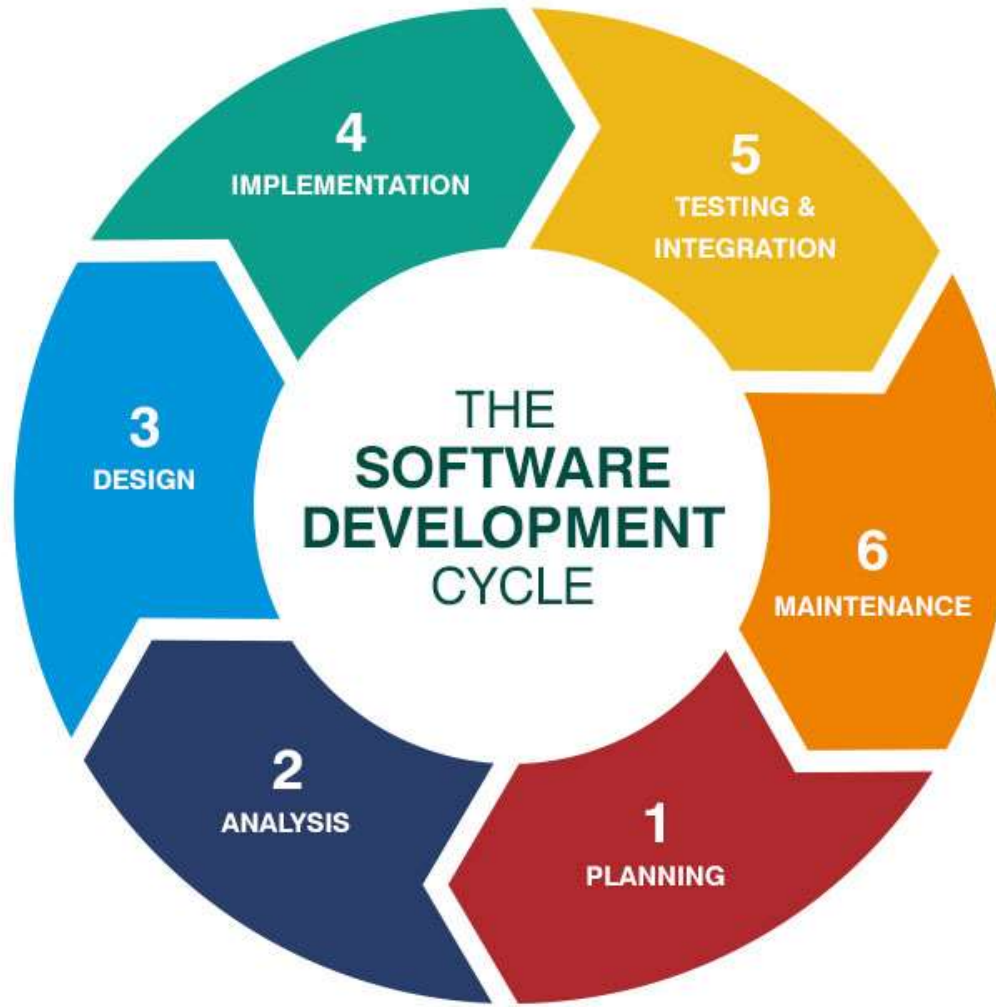
- از سه فاز اصلی و فازهای چتری (پشتیان) تشکیل شده است.

- فاز تعریف

- فاز توسعه

- فاز نگهداری

# چرخه حیات توسعه نرم افزار (Software Development Lifecycle)



# فاز تعریف

■ تاکید بر چه (WHAT) است:

— چه عملکردی انتظار می رود؟

— سیستم چه رفتاری باید داشته باشد؟

— چه واسطه‌های کاربری‌ای باید طراحی شود؟

— چه محدودیت‌هایی برای طراحی وجود دارد؟

— معیارهای صحت اعتبار برای تعریف یک نرم افزار موفق چیست؟

■ در این مرحله روش مورد استفاده با انجام سه فعالیت زیر مشخص می‌شود:

— مهندسی سیستم یا اطلاعات (System/Information Engineering)

— برنامه‌ریزی پروژه نرم افزاری (Software Project Planning)

— تحلیل نیازمندی‌ها (Requirement's Analysis)

- بر چگونگی (HOW) تاکید دارد:
  - چگونه داده‌ها باید سازمان دهی شوند؟
  - تکمیل وظایف چگونه در معماری وظایف گنجانده می‌شود؟
  - ارتباط و توصیف پردازش‌ها چگونه انجام می‌گیرد؟
  - ویژگی‌های واسط‌های کاربری چیست؟
  - چگونه آزمایش انجام می‌گیرد؟
  - چگونه طراحی تبدیل به معماری برنامه و کد می‌شود؟ (مهمترین فعالیت)
- در این مرحله مدل‌های مورد نیاز تولید می‌گردد و روند کلی برنامه‌ها تعیین می‌شود. سه فعالیت در این مرحله عبارتند از:
  - طراحی نرم افزار (Software Design)
  - تولید کد (Coding)
  - تست (Testing)

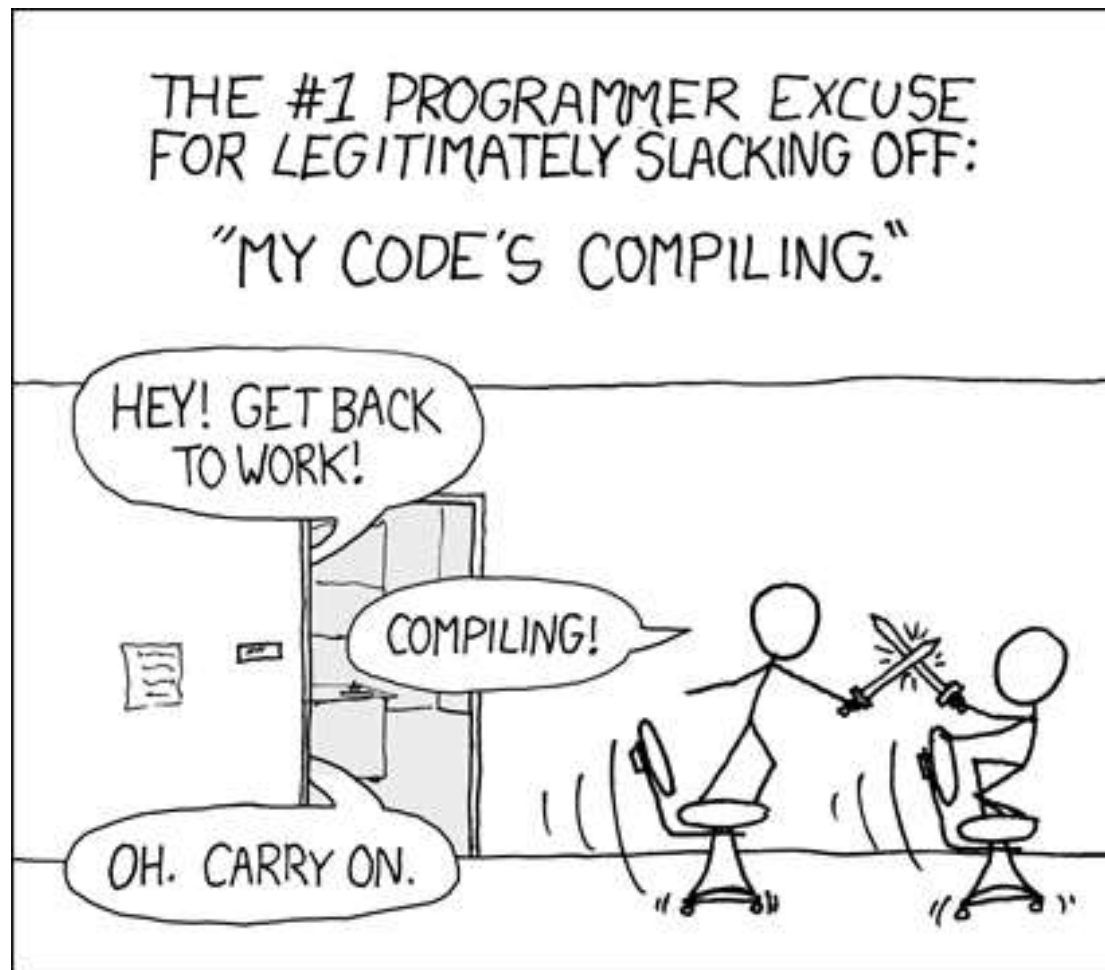
# فاز نگهداری

- نگهداری اصلاحی (Correction): اشکالات در چرخه زندگی پروژه شناسایی، بررسی و رفع می‌گردند.
- نگهداری تطبیقی (Adaptation): تغییر به مرور زمان و با توجه به نیازهای محیطی و کاربر. نیازهای جدید را باید با محیط تطبیق داد منوط به اینکه نرم افزار قابلیت توسعه را داشته باشد.
- نگهداری ارتقایی (Enhancement): ارتقا نرم افزار پس از راه اندازی که توسط کاربر خواسته می‌شود.
- نگهداری پیشگیرانه (Prevention): ایجاد تغییراتی در برنامه‌ها که اصلاح و بهبود نرم افزار را تسهیل می‌کند. نرم افزار پس از دوره مشخصی بازنگری شده و اصلاحات لازم انجام می‌شود، در واقع نرم افزار جدیدی تولید می‌گردد.

# فعالیت‌های چتری (پشتیبانی)

■ فعالیت‌های پشتیبانی زیر در کلیه مراحل انجام می‌شود:

- کنترل و نظارت بر برنامه‌ریزی پروژه
- بازنگری‌های مستمر فنی و رسمی (Quality Control)
- اطمینان از مرغوبیت نرم افزار (Quality Assurance)
- مدیریت پیکربندی نرم افزار (Configuration/Change Management)
- تهیه و تدوین مستندات
- اندازه‌گیری (Software Measurements)
- مدیریت پیامدهایی که برآورد زمان و هزینه را تغییر دهد (Risk Management)



چرخه حیات نرم افزار باید مدیریت شود!



## فرایند توسعه نرم افزار (Software Engineering Process)

- به مجموعه‌ای از فعالیت‌های مهندسی اطلاق می‌شود که با هدف مدیریت چرخه حیات یک محصول نرم‌افزاری، طراحی و برنامه‌ریزی می‌شود.
- فرایندهای تولید نرم‌افزار با این هدف بوجود آمده‌اند که این مجموعه فعالیت‌ها را در یک قالب مشخص ساماندهی، استاندارد و مستند کرده و به این ترتیب سرعت و کیفیت تولید نرم‌افزار را بهبود بخشند.

# فرایند توسعه نرم افزار

- فرایندهای تولید نرم افزار بر اساس مدل تولید نرم افزار را می توان به چند دسته تقسیم کرد. از جمله:

– توسعه ترتیبی (Sequential Development)

- مدل آبشاری (Waterfall Model)

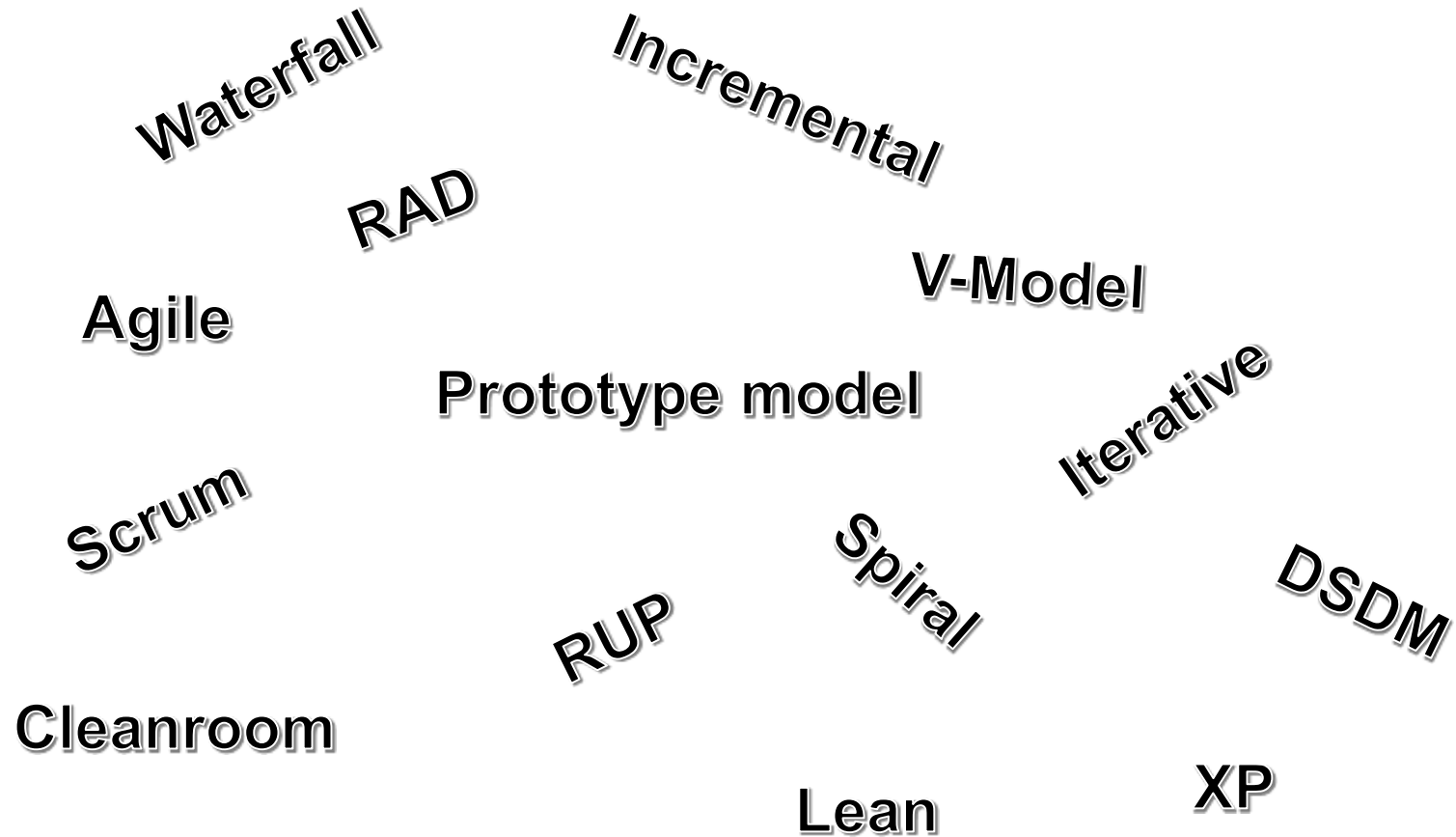
- مدل حلزونی (Spiral Model)

– توسعه تکرار شونده و افزایشی ( Iterative and Incremental )

(Development

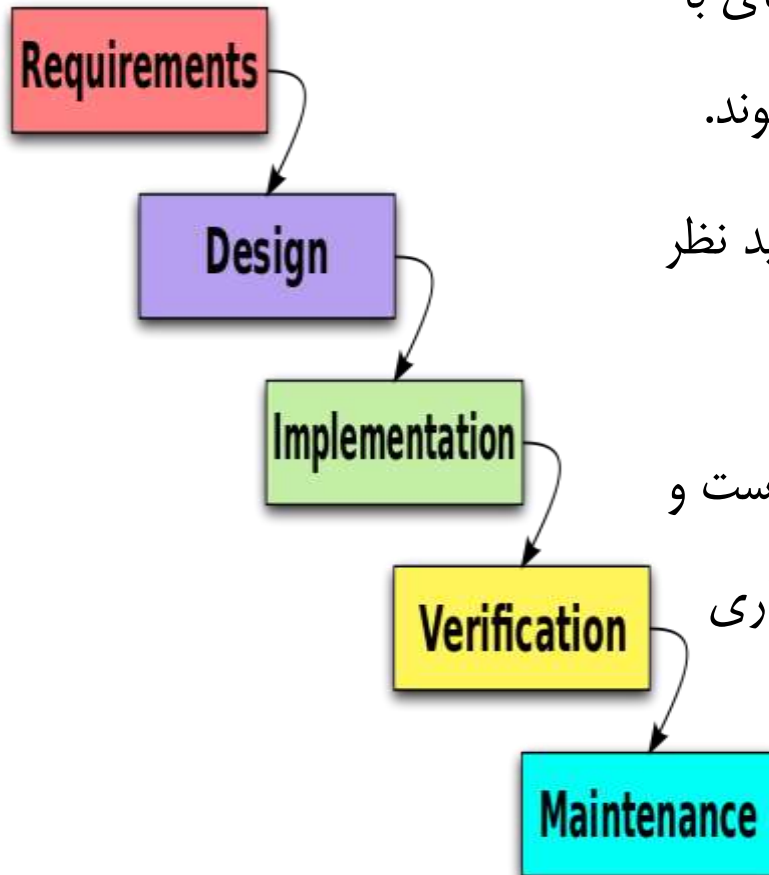
– توسعه چابک (Agile Development)

# فرایند توسعه نرم افزار



# توسعه ترتیبی نرم افزار (Sequential Development)

## ■ مدل آبشاری (Waterfall)



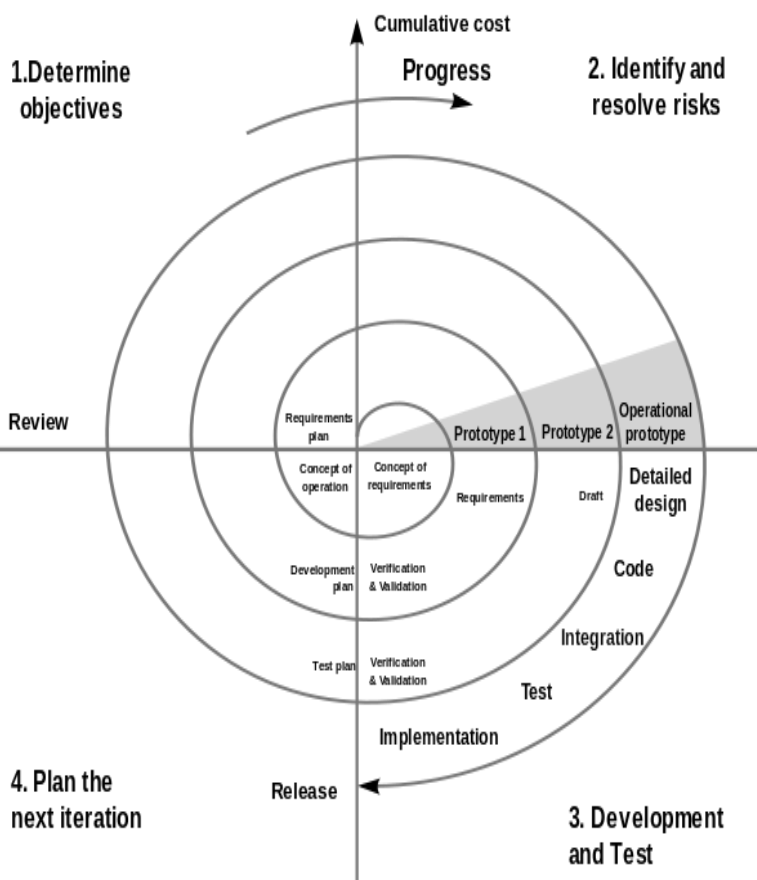
- در این مدل فعالیت‌های تولید نرم‌افزار در قالب فازهای با توالی مشخص و به ترتیب، برنامه‌ریزی و اجرا می‌شوند.
- اشکال عمده این روش این است که بازبینی و تجدید نظر در فازهای انجام شده امکان پذیر نیست.
- خطای تخمین ابعاد پروژه، ریسک اشتباه در فهم درست و تحلیل نیازمندی‌ها و نیز امکان انتخاب نایجای معماری بسیار بالا می‌باشد.

# توسعه ترقیبی نرم افزار (Sequential Development)

## ■ مدل حلزونی (Spiral Model)

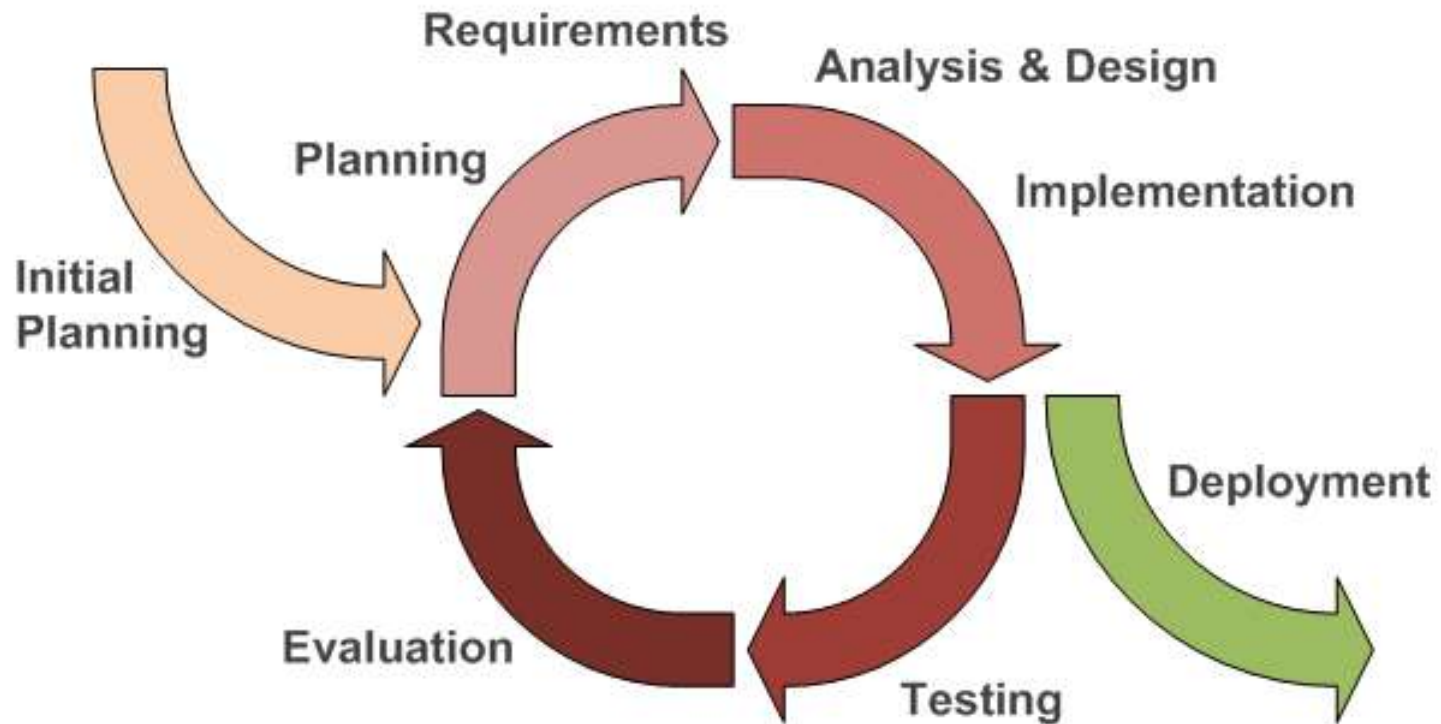
— این مدل برای بهبود مدیریت ریسک در تمام مراحل تولید نرم افزار بوجود آمد و ترکیبی بود از مدل آبشاری به همراه امکان نمونه سازی (Prototyping) سریع.

— این مدل اگر چه برخی جنبه های مدل آبشاری را بهبود می داد لیکن از نارسایی های فراوان آن مدل متاثر بود.



# توسعه تکرار شونده و افزایشی

- Iterative and Incremental Development



# توسعه تکرار شونده و افزایشی

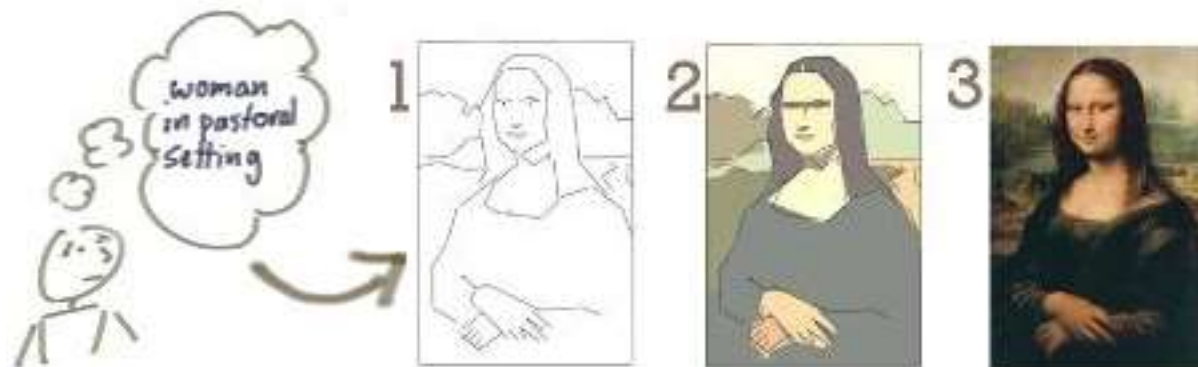
- به وسیله این روش می توان روی قسمت های مهمتر پروژه متمرکز شد و از پرداختن به قسمت های کم اهمیت تر (در هر تکرار) پرهیز نمود.
- آزمایش تکراری و مستمر امکان تشخیص بهتر روند پیشرفت پروژه را به ما می دهد.
- باعث می شود بار کاری (Workload) تیم ها روی چرخه تولید پروژه به صورت متوازن توزیع شود.
- خطاهایی که در درک مسئله، تحلیل یا طراحی رخ داده است زودتر تشخیص داده می شود.
- ناسازگاری های بین نیازمندی ها، تحلیل، طراحی و پیاده سازی زودتر تشخیص داده می شود.
- ذینفعان می توانند دائما از روند پیشرفت پروژه مطلع گردند.

# توسعه تکرار شونده و افزایشی

## Incremental



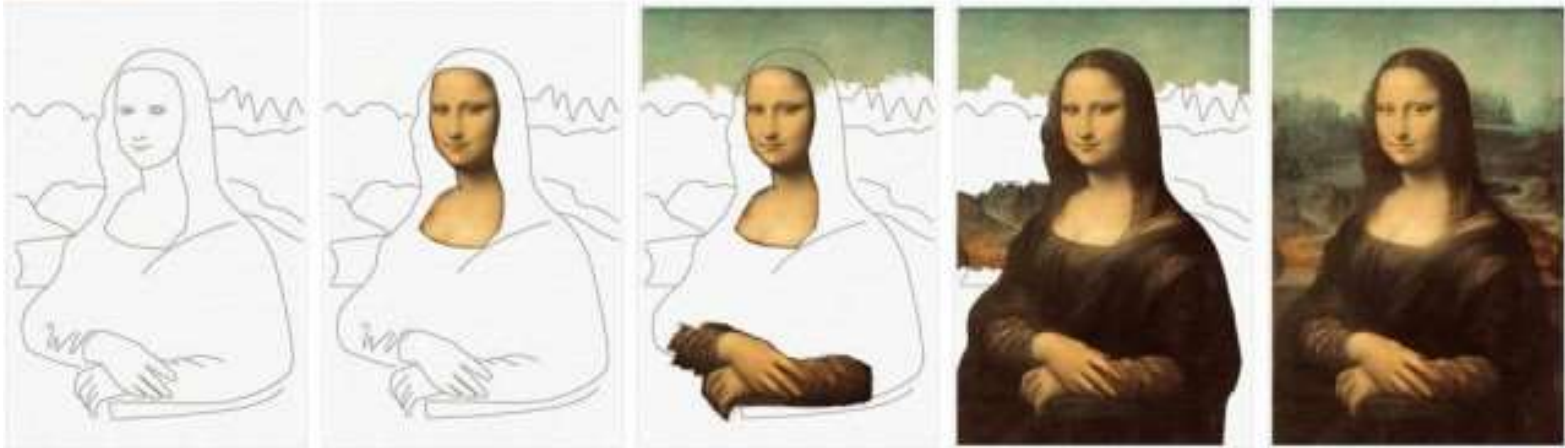
## Iterative





# توسعه تکرار شونده و افزایشی

JONAH



# توسعه چابک (Agile Development)

- Agile مجموعه‌ای از ارزش‌ها و اصول جهت توسعه نرم‌افزارهای کارا توسط تیم‌های خود سازمانده است.
- ارزش‌ها و اصول Agile در سال ۲۰۰۱ توسط ۱۷ نفر از اساتید صنعت توسعه نرم‌افزار طی یک بیانیه با عنوان بیانیه توسعه چابک تنظیم و ارائه شده است.
- اساس و هدف این اصول و ارزش‌ها ارائه نرم‌افزار کارا و یا محصول کارا به مشتری می‌باشد.

# ارزش های توسعه نرم افزار چابک

**افراد و تعاملات** بالاتر از فرآیندها و ابزارها

**نرم افزار کارا** بالاتر از مستند سازی جامع

**همکاری مشتری** بالاتر از قرارداد کار

**جوابگویی به تغییرات** بالاتر از پیروی از یک طرح

با وجود اینکه در موارد سمت چپ ارزش‌هایی وجود دارد ولی موارد سمت راست ارزش بیشتری برای ما دارند

# ارزش های توسعه نرم افزار چابک

- **افراد و تعاملات** به معنی ایجاد یک محیط پویا و همکارانه بین اعضای تیم.
- **همکاری مشتری** به معنی دستیابی به فیدبک های مشتری و امکان ایجاد نوآوری.
- **جوابگویی به تغییرات** در جهت دستیابی به ارزش نرم افزار کارا.
- به عبارت ساده تر به دنبال تیمی با تعاملات خوب بین اعضای آن هستیم که این تیم با گرفتن فیدبک های مشتری به صورت متوالی و سازگار کردن محصول با نیازهای مشتری در حال توسعه نرم افزار به صورت کارا است.

# اصول بیانیه چابک

- بالاترین اولویت ما رضایت مشتری از طریق تحویل به موقع و مداوم نرم افزار ارزشمند می باشد.
- پذیرائی از نیازهای در حال تغییر، حتی آنهایی که در اواخر توسعه پدید آور می شوند. فرآیندهای چابک تغییرات را جهت رقابت بر سر مشتری مهار و کنترل می نمایند.
- تحویل نرم افزار کار کننده غالباً از چند هفته تا چند ماه یک بار انجام می شود که زمانبندی کوتاه تر ترجیح داده می شود.
- ذینفعان تجاری و توسعه دهندگان باید هر روزه در طول پروژه با هم کار کنند.

# اصول بیانیه چابک

- پروژه‌ها را بر روی افراد با انگیزه بنا کنید. محیط لازم را به آنها بدهید و از نیازهای آنها پشتیبانی نمایید و به آنها اعتماد نمایید تا کارها را انجام بدهند.
- کارآمدترین و موثرترین روش برای انتقال و رساندن اطلاعات به تیم توسعه، گفتگوی چهره به چهره و رو در رو می‌باشد.
- نرم افزار کار کننده اصلی‌ترین معیار پیشرفت می‌باشد.
- فرآیندهای چابک توسعه پایدار را ترویج می‌دهند. حامیان مالی، توسعه دهندگان و کاربران باید قادر به حفظ سرعت پیشرفت ثابتی برای یک مدت نامحدود باشند.

# اصول بیانیه چابک

- توجه مداوم به برتری فنی و طراحی خوب باعث افزایش چابکی می شود.
- اصل سادگی ضروری می باشد.
- بهترین معماری ها، نیازمندی ها و طراحی ها از تیم های خود سازمانده پدیدآور می شود.
- در فواصل منظم، تیم بر چگونگی موثرتر شدن تامل و تفکر می نماید و سپس تیم رفتار خود را بر اساس بازتاب این تفکر تنظیم و هم سو می نماید.

# فرایند توسعه نرم افزار

■ فرایند توسعه نرم افزار به وسیله‌ی **متدلوژی‌ها** یا **چارچوب‌های** توسعه نرم‌افزار پیاده‌سازی می‌شوند. مانند:

— توسعه ترتیبی (مدل آبشاری):

- Structured systems analysis and design method (SSADM)

— توسعه تکرار شونده و افزایشی:

- RUP Methodology

■ توسعه چابک:

- SCRUM Framework



- متدولوژی مجموعه‌ی روش‌ها و خط‌مشی‌های گام به گام است که برای تکمیل یک یا چند مرحله از مراحل یک چرخه تکاملی به کار گرفته می‌شود.
- هر روش تکنیک‌ها و استانداردهای خاص خود را به چرخه تکاملی تحمیل می‌کند.

- چارچوب جزئیات دقیق و کاملی در مورد اینکه کارها در پروژه چگونه باید انجام شوند ارائه نمی‌دهد و آنرا به تیم‌ها واگذار می‌کند.
- کشف راه حل برای مشکلات به تیم‌ها واگذار می‌شود و نسخه‌های پیچیده شده و لازم الاجرا وجود ندارد.

## متدلوژی‌های توسعه نرم‌افزار (SDM: Software Development Methodology)

- راهکاری برای استفاده از روش‌های مهندسی نرم‌افزار با هدف ارائه ابزار لازم برای توسعه سیستم‌های نرم‌افزاری متمرکز.
- عمدتاً متشکل از:
  - مجموعه‌ای از قراردادهای مدل‌سازی متشکل از یک زبان مدل‌سازی (نحو و معنانشناسی)
  - یک فرایند مهندسی نرم‌افزار

## متدلوژی‌های توسعه نرم‌افزار (SDM: Software Development Methodology)

■ یک متدلوژی توسعه نرم‌افزار مشخص می‌کند:

— چگونه نیازمندی‌ها باید به نرم‌افزار تبدیل شوند.

— چه کسی (who)، چه چیزی (what)، چه زمانی (when)، چرا (why) و چگونه

(how) در توسعه نرم‌افزار را مشخص می‌کند.

## متدلوژی‌های توسعه نرم‌افزار (SDM: Software Development Methodology)

### ■ متدلوژی توسعه نرم‌افزار چهار نقش اساسی دارد:

- مشخص نمودن ترتیب انجام فعالیت‌ها برای تبدیل نیازمندی‌های ذینفعان به محصول.
- مشخص نمودن دستاوردها/فراآورده‌هایی (artifacts) که باید توسعه یابند (با استفاده از زبان مدل‌سازی).
- تعیین روش اداره وظایف توسعه دهندگان را به شکل فردی و تیم به شکل کلی و انتساب نقش‌ها به افراد.
- تعیین معیارهای نظارت و اندازه‌گیری کیفیت محصولات و روند پیشرفت فعالیت‌های پروژه.

## ■ چرخه حیات توسعه نرم افزار

– فرایند توسعه نرم افزار

■ توسعه ترتیبی نرم افزار

– مدل آبشاری

» SSADM

– مدل حلزونی

■ توسعه تکرار شونده و افزایشی

– متدلوژی RUP

■ توسعه چابک

– چارچوب اسکرام

# چرخه توسعه نرم افزار در واقعیت

