

# مهندسی نرم افزار

## (متودولوژی)

نگارنده: آرش زار عیان جهرمی

مدرس: محمد احمدزاده

رشته مهندسی حرفه ای کامپیوتر

دانشکده فنی میناب

# ◆ METHODOLOGIEST ◆

در دنیای توسعه نرم افزار، متدولوژی‌های مختلفی برای مدیریت پروژه وجود دارند که هر یک از آن‌ها دارای مزایا، معایب و کاربردهای خاص خود هستند. در اینجا به مقایسه‌ای میان XP، Scrum، AUP (Agile)، RUP (Rational Unified Process)، Kanban، (Extreme Programming)، Unified Process، DSDM (Dynamic Systems Development Method) و DevOps می‌پردازیم.

## 1. اسکرام (Scrum)

اسکرام یکی از محبوب‌ترین چارچوب‌های چابک (Agile) برای مدیریت پروژه است. این متدولوژی مبتنی بر ایجاد چرخه‌های کاری کوتاه به نام اسپرینت‌ها است که معمولاً بین 1 تا 4 هفته طول می‌کشد. در هر اسپرینت، تیم تلاش می‌کند تا ویژگی‌های مشخصی را به پایان برساند و یک نسخه کاری از محصول را ارائه دهد.

## عناصر اصلی اسکرام:

تیم خودمدیریت: تیم‌ها بدون نیاز به نظارت سنگین مدیریت کار می‌کنند.

جلسات روزانه (Daily Standups): تیم هر روز جلسه‌ای کوتاه دارد تا وضعیت کار را بررسی کند.

اسپرینت‌های کوتاه: هر چرخه زمانی کوتاه و با هدف ارائه یک محصول قابل کار در پایان است.

نقش‌ها: شامل مالک محصول (Product Owner)، اسکرام‌مستر (Scrum Master) و تیم توسعه است.

## مزایا:

تعامل و همکاری تیمی بالا: جلسات روزانه و بازبینی‌های دوره‌ای.

سازگاری با تغییرات: انعطاف‌پذیری در مواجهه با تغییر نیازها.

تحويل زودهنگام: قابلیت ارائه نسخه‌های قابل استفاده در پایان هر اسپرینت.

## معایب:

نیاز به تعهد بالا: تیم باید به جلسات روزانه و اسپرینت‌ها متعهد باشد.

پیچیدگی مدیریت: ممکن است نیاز به یک اسکرام‌مستر ماهر باشد.

نامناسب برای پروژه‌های بزرگ‌مقیاس: نیاز به تقسیم پروژه به چندین تیم کوچک دارد.

## کاربرد:

مناسب برای پروژه‌های کوچک تا متوسط با تیم‌های خودمدیریت و تعاملی که نیاز به تحويل سریع و انعطاف‌پذیری بالا دارند.

## 2. برنامه‌نویسی افراطی (XP - Extreme Programming)

XP یکی دیگر از روش‌های چابک است که به طور ویژه بر بهبود کیفیت نرم‌افزار و پاسخ سریع به تغییرات تأکید دارد. XP شامل مجموعه‌ای از بهترین شیوه‌های توسعه نرم‌افزار است که تعامل نزدیک بین تیم توسعه و مشتری و تکرار سریع چرخه‌های توسعه را تشویق می‌کند.

## **عناصر اصلی XP:**

برنامه‌نویسی زوجی (Pair Programming): دو نفر به صورت مشترک کدنویسی می‌کنند.

توسعه تست‌محور (TDD): نوشتن تست قبل از کد برای بهبود کیفیت.

انتشار مکرر: ارائه مداوم به مشتری در بازه‌های زمانی کوتاه.

طراحی ساده: طراحی کد به ساده‌ترین شکل ممکن.

## **مزایا:**

تمرکز بر کیفیت کد: از طریق تکنیک‌های مانند برنامه‌نویسی زوجی و تست‌های واحد.

تعامل مستمر با مشتری: نیازهای مشتری به طور مداوم بررسی می‌شوند.

تحويل مداوم: امکان تحويل کد به طور مداوم و در فواصل زمانی کوتاه.

## **معایب:**

فشار بالا روی توسعه‌دهندگان: نیاز به تحويل مداوم کد.

تست زیاد: تست‌های مداوم ممکن است زمان‌بر باشد.

نامناسب برای تیم‌های بزرگ: برای تیم‌های بزرگ ممکن است هماهنگی چالش‌برانگیز باشد.

## **کاربرد:**

مناسب برای پروژه‌های با تغییرات سریع و نیاز به کیفیت کد بالا. اغلب در پروژه‌های کوچک تا

متوسط استفاده می‌شود.

### 3. کانبان (Kanban)

کانبان یک روش مدیریت جریان کار است که ریشه در سیستم تولیدی تویوتا دارد. این روش بیشتر برای مدیریت پروژه‌های جاری و فرآیندهای تکرارشونده مناسب است. کانبان از تخته‌های بصری (کانبان بورد) استفاده می‌کند تا مراحل و وضعیت کارهای در دست اجرا را نمایش دهد.

#### عناصر اصلی کانبان:

بصری‌سازی کار: استفاده از بردهایی برای نمایش وضعیت وظایف.  
محدودیت‌های WIP (کارهای در جریان): محدود کردن تعداد وظایفی که همزمان انجام می‌شود.  
مدیریت پیوسته جریان: تلاش برای بهبود مستمر جریان کار.

#### مزایا:

سادگی و انعطاف‌پذیری: بدون محدودیت‌های خاص بر چرخه‌های زمانی (اسپرینت‌ها).  
بصری‌سازی جریان کار: امکان مدیریت بهتری از جریان کار و وظایف در دست انجام.  
بهبود مستمر: امکان شناسایی گلوگاه‌ها و بهبود کارایی.

#### معایب:

نبود زمان‌بندی مشخص: ممکن است تحویل پروژه به تعویق بیفتد.  
عدم وجود چارچوب دقیق: ممکن است برای تیم‌های بزرگ و پیچیده چالش‌برانگیز باشد.

## کاربرد:

مناسب برای پروژه‌های جاری با نیاز به انعطاف‌پذیری و مدیریت جریان کاری بدون نیاز به زمان‌بندی ثابت.

### 4. فرآیند یکپارچه رشنال (RUP - Rational Unified Process)

RUP (فرآیند یکپارچه‌ی رشنال) یک چارچوب فرآیندی است که توسط شرکت IBM توسعه یافته است و روشی سیستماتیک برای مدیریت و اجرای پروژه‌های نرم‌افزاری ارائه می‌دهد. این روش بر اساس مراحل مختلف توسعه شامل تعریف نیازها، طراحی، توسعه، آزمایش و نگهداری است و به صورت فازبندی شده پروژه را مدیریت می‌کند.

### عناصر اصلی RUP:

فازبندی پروژه: شامل چهار فاز (آغاز، توسعه، ساخت و انتقال).  
مستندسازی قوی: تأکید بر تهیه مستندات در هر مرحله از پروژه.  
تعامل با مشتری: در هر فاز، نیازهای مشتری بررسی و تأیید می‌شود.

### مزایا:

ساختار دقیق و کامل: چهار فاز (شروع، توسعه، ساخت و نگهداری) به تیم کمک می‌کند تا در تمام مراحل پروژه هدایت شود.  
پشتیبانی قوی از مستندات: تمرکز بر تهیه مستندات جامع و دقیق.

پروژه‌های بزرگ مقیاس: مناسب برای پروژه‌های پیچیده و بزرگ.

### معایب:

پیچیدگی و سنگینی فرآیند: اجرای کامل RUP می‌تواند زمان‌بر و پیچیده باشد.  
سازگاری کمتر با تغییرات: ممکن است برای پروژه‌هایی که به تغییرات مکرر نیاز دارند مناسب نباشد.

### کاربرد:

مناسب برای پروژه‌های بزرگ و پیچیده که نیاز به مستندات جامع و فرآیندهای دقیق دارند.

## 5. فرآیند یکپارچه چابک (AUP - Agile Unified Process)

AUP (فرآیند یکپارچه چابک) نسخه چابک شده‌ی RUP است که به دنبال کاهش پیچیدگی و افزودن انعطاف‌پذیری به فرآیند سنتی RUP است. در AUP، تمرکز بیشتری بر روی تعاملات تیمی و تغییرات سریع قرار دارد، در حالی که ساختار و فازبندی RUP حفظ شده است.

### عناصر اصلی AUP:

فازبندی پروژه: مانند RUP فازبندی شده اما با انعطاف بیشتر.  
تمرکز بر چابکی: کاهش مستندات و استفاده از روش‌های چابک برای پاسخ به تغییرات.  
تمرکز بر تعامل تیمی: همانند متدولوژی‌های چابک، تعامل مستمر بین تیم توسعه و مشتری وجود دارد.

## **مزایا:**

**انعطاف‌پذیری و چابکی:** ترکیب اصول چابک با فرآیند RUP.

**سازگاری با تغییرات:** امکان سازگاری بهتر با تغییرات نسبت به RUP.

**فازبندی مناسب:** شامل فازهای تعیین‌شده‌ای که در هر مرحله از پروژه مفید است.

## **معایب:**

**سنگینی بیشتر نسبت به سایر روش‌های چابک:** هنوز نیاز به مستندات و فرآیندهای جامع دارد.

**پیچیدگی برای تیم‌های کوچک:** ممکن است برای تیم‌های کوچک و ساده بیش از حد پیچیده باشد.

## **کاربرد:**

**مناسب برای پروژه‌هایی که نیاز به ترکیبی از مستندات دقیق و انعطاف‌پذیری چابک دارند.**

## **6. روش توسعه سیستم‌های پویا ( DSDM - Dynamic Systems Development Method )**

DSDM یکی از روش‌های قدیمی توسعه چابک است که بر روی تحویل سریع و مشارکت قوی با مشتری تمرکز دارد. این متدولوژی تأکید ویژه‌ای بر تعاملات تیم و مشتری و همچنین هم‌ترازی توسعه با نیازهای کسب‌وکار دارد. DSDM از چرخه‌های تکراری و افزایشی استفاده می‌کند و به شدت بر تحویل کارکردهای کلیدی تأکید دارد.



## **عناصر اصلی DSDM:**

**اولویت‌دهی به نیازها: تأکید بر نیازهای مهم کسب‌وکار.**

**چرخه‌های توسعه کوتاه: مشابه سایر متدهای چابک، تحویل سریع ویژگی‌های اصلی.**

**تمرکز بر کیفیت: تضمین کیفیت از طریق بازبینی‌های منظم.**

## **مزایا:**

**توجه به نیازهای تجاری: تأکید بر اینکه توسعه نرم‌افزار باید با نیازهای تجاری هماهنگ باشد.**

**تحویل سریع و به‌موقع: تأکید بر ارائه سریع و مرتب.**

**تعامل مستمر با مشتری: نیازهای مشتری به طور مداوم در نظر گرفته می‌شود.**

## **معایب:**

**نیاز به تعهد تیم: تیم باید به روش‌های DSDM کاملاً متعهد باشد.**

**پیچیدگی در پیاده‌سازی: اجرای صحیح DSDM ممکن است نیاز به منابع و زمان بیشتری داشته باشد.**

## **کاربرد:**

**مناسب برای پروژه‌های با نیازهای تجاری مشخص که به سرعت تحویل و همکاری با مشتری نیاز دارند.**

## 7. دو آپس (DevOps)

دو آپس (DevOps) یک فلسفه و روش کاری است که بر ادغام فرآیندهای توسعه (Development) و عملیات (Operations) تأکید دارد. هدف اصلی دو آپس افزایش همکاری بین تیمهای توسعه و عملیات، بهبود سرعت تحویل و تضمین کیفیت نرم افزار است. این روش از اتوماسیون برای تسریع فرآیندهای تست، تحویل و استقرار استفاده می کند.

### عناصر اصلی DevOps:

یکپارچگی و همکاری تیمها: توسعه و عملیات به صورت یکپارچه کار می کنند.  
تحویل مداوم (Continuous Delivery): استفاده از اتوماسیون برای استقرار سریع و مداوم.  
نظارت و بازخورد مداوم: نظارت بر عملکرد سیستمها و بهبود آنها در چرخه های مکرر.

### مزایا:

افزایش سرعت توسعه و تحویل: از طریق اتوماسیون و همکاری تیمهای توسعه و عملیات.  
بهبود کیفیت نرم افزار: کاهش خطاها و مشکلات در فرآیند تحویل.  
نظارت مداوم: امکان نظارت و بهبود مستمر پس از تحویل.

### معایب:

پیچیدگی در پیاده سازی: نیاز به فرهنگ سازمانی مناسب و ابزارهای پیشرفته.  
نیاز به تخصص بالا: تیمها باید در اتوماسیون، نظارت و ابزارهای DevOps تخصص داشته باشند.

## کاربرد:

مناسب برای پروژه‌هایی که به سرعت توسعه و تحویل مداوم نیاز دارند، به خصوص در محیط‌های ابری و سیستم‌های بزرگ.

در کل روش‌هایی مانند اسکرام و XP بیشتر بر چابکی و انعطاف‌پذیری در چرخه‌های کوتاه و تعاملات مستمر با مشتری متمرکز هستند. در مقابل، RUP و DSDM برای پروژه‌های بزرگ و پیچیده مناسب‌ترند، جایی که مستندسازی و مدیریت دقیق لازم است. کانبان و AUP به عنوان روش‌های میانی، ترکیبی از انعطاف‌پذیری و ساختار را ارائه می‌دهند. بسته به اندازه پروژه، نیازهای تیم و سطح انعطاف‌پذیری مورد انتظار، هر یک از این روش‌ها می‌توانند انتخاب مناسبی باشند.

روش	تعریف	مزایا	معایب	کاربرد
Scrum	چارچوب چابک که بر اسپرینت‌های کوتاه تمرکز دارد	- تعامل و همکاری بالا - سازگاری با تغییرات - تحویل سریع و مداوم	- مناسب نبودن برای پروژه‌های پیچیده و بلندمدت - نیاز به تیم خودمدیریت	پروژه‌های کوتاه‌مدت یا متوسط با نیاز به تعامل زیاد و تحویل سریع
XP	روش چابک با تمرکز بر کیفیت کد و توسعه سریع با تکنیک‌هایی مانند برنامه‌نویسی زوجی و TDD	- کیفیت بالای کد - تعامل نزدیک با مشتری - تحویل سریع و مداوم	- هزینه بالای برنامه‌نویسی زوجی - فرسودگی تیم	پروژه‌های نرم‌افزاری که نیاز به کیفیت بالا و پاسخ سریع به تغییرات دارند

Kanban	<p>روش بصری سازی جریان کار با تمرکز بر کاهش کارهای در جریان و بهبود کارایی.</p>	<p>- بصری سازی ساده جریان کار - انعطاف پذیری بالا - بهبود مستمر</p>	<p>- عدم وجود زمان بندی مشخص - پیچیدگی مدیریت کارها</p>	<p>پروژه هایی که به انعطاف پذیری بالا و مدیریت مستمر کار نیاز دارند</p>
RUP	<p>چارچوب فرآیندی ساختاریافته با فازهای مختلف برای توسعه نرم افزار، با تأکید بر مستندسازی و مدیریت پروژه.</p>	<p>- ساختار دقیق و جامع - کاهش ریسک - توسعه تدریجی و افزایشی</p>	<p>- پیچیدگی و سنگینی فرآیند - کاهش چابکی - نیاز به منابع زیاد</p>	<p>پروژه های بزرگ و پیچیده با نیاز به مستندسازی جامع و مدیریت دقیق</p>
DSDM	<p>روش چابک با تمرکز بر تحویل سریع و اولویت بندی نیازهای تجاری.</p>	<p>- تمرکز بر نیازهای تجاری - تحویل سریع - تعامل مداوم با مشتری</p>	<p>- هزینه بالای پیاده سازی - پیچیدگی در پروژه های کوچک</p>	<p>پروژه های تجاری پیچیده با نیاز به تحویل سریع و تعامل مداوم</p>
DevOps	<p>ترکیبی از توسعه و عملیات با هدف خودکارسازی و همکاری برای تحویل سریع و پایدار نرم افزار.</p>	<p>- تحویل مداوم و سریع تر - همکاری بهتر بین تیم ها - پایداری بیشتر نرم افزار</p>	<p>- نیاز به تغییرات فرهنگی و سازمانی - پیچیدگی در پیاده سازی</p>	<p>پروژه هایی که به تحویل سریع، خودکارسازی فرآیندها و پایداری نیاز دارند</p>