

# METHODOLOGIES

علیرضا حسن زاده  
دانشگاه فنی حرفه ای میناب

به نام خدا

درس: مهندسی نرم افزار

کارشناسی پیوسته

مدرس: محمد احمد زاده

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 **Scrum**، **XP (Extreme Programming)**، **Kanban**، **AUP (Agile Unified Process)**، **RUP (Rational Unified Process)**، **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.

### • مزایا:

تعامل و همکاری تیمی بالا: جلسات روزانه و بازبینی‌های دوره‌ای. سازگاری با تغییرات: انعطاف‌پذیری در مواجهه با تغییر نیازها. تحویل 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 تحویل 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 نرم افزار و پاسخ سریع به تغییرات تأکید دارد. XP شامل مجموعه‌ای از بهترین شیوه‌های توسعه نرم افزار است که تعامل نزدیک بین تیم توسعه و مشتری و تکرار سریع چرخه‌های توسعه را تشویق می‌کند.

### • 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.

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

- معایب:

فشار بالا روی توسعه‌دهندگان: نیاز به تحویل 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.

- 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. UP (Unified Process):** یک چارچوب توسعه نرم‌افزار است که شامل مراحل مانند تحلیل، طراحی، پیاده‌سازی و تست است. هر مرحله به تکرارهای کوتاه تقسیم می‌شود تا ریسک‌ها کاهش یابد و کیفیت بهبود یابد. UP به تیم‌ها کمک می‌کند تا با یک رویکرد ساختارمند، پروژه‌ها را به صورت تدریجی و افزایشی توسعه دهند.

- مزایا:

- مدیریت ریسک: با استفاده از تکرارهای کوتاه، ریسک‌های پروژه به‌طور مؤثر مدیریت می‌شود.
- کیفیت بالا: تمرکز بر تحلیل دقیق و طراحی قبل از پیاده‌سازی به بهبود کیفیت کمک می‌کند.
- مناسب برای پروژه‌های بزرگ: ساختارمندی UP آن را برای پروژه‌های بزرگ و پیچیده مناسب می‌کند.
- معایب:
- نیاز به مستندسازی زیاد: این روش نیازمند مستندسازی جامع و دقیق است که می‌تواند زمان‌بر باشد.
- کمبود انعطاف‌پذیری: در مقایسه با روش‌های سریع مانند اسکرام، UP انعطاف‌پذیری کمتری دارد.
- پیاده‌سازی پیچیده: نیازمند مهارت‌ها و دانش فنی بالایی برای پیاده‌سازی موفق است.

- **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.

- 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.

- 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. دواپس (DevOps)

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

- مزایا:

افزایش سرعت توسعه و تحویل: از طریق اتوماسیون و همکاری تیم‌های توسعه و عملیات.

بهبود کیفیت نرم‌افزار: کاهش خطاها و مشکلات در فرآیند تحویل.  
نظارت مداوم: امکان نظارت و بهبود مستمر پس از تحویل.

- معایب:

پیچیدگی در پیاده‌سازی: 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.

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	روش بصری‌سازی جریان کار با تمرکز بر کاهش کارهای در جریان و بهبود کارایی.	- بصری‌سازی ساده جریان کار - انعطاف‌پذیری بالا - بهبود مستمر	- عدم وجود زمان‌بندی مشخص - پیچیدگی مدیریت کارها	پروژه‌هایی که به انعطاف‌پذیری بالا و مدیریت مستمر کار نیاز دارند
RUP	چارچوب فرآیندی ساختاریافته با فازهای مختلف برای توسعه نرم‌افزار، با تأکید بر مستندسازی و مدیریت پروژه.	- ساختار دقیق و جامع - کاهش ریسک - توسعه تدریجی و افزایشی	- پیچیدگی و سنگینی فرآیند - کاهش چابکی - نیاز به منابع زیاد	پروژه‌های بزرگ و پیچیده با نیاز به مستندسازی جامع و مدیریت دقیق
DSDM	روش چابک با تمرکز بر تحویل سریع و اولویت‌بندی نیازهای تجاری.	- تمرکز بر نیازهای تجاری - تحویل سریع - تعامل مداوم با مشتری	- هزینه بالای پیاده‌سازی - پیچیدگی در پروژه‌های کوچک	پروژه‌های تجاری پیچیده با نیاز به تحویل سریع و تعامل مداوم
DevOps	ترکیبی از توسعه و عملیات با هدف خودکارسازی و همکاری برای تحویل سریع و پایدار نرم‌افزار.	- تحویل مداوم و سریع‌تر - همکاری بهتر بین تیم‌ها - پایداری بیشتر نرم‌افزار	- نیاز به تغییرات فرهنگی و سازمانی - پیچیدگی در پیاده‌سازی	پروژه‌هایی که به تحویل سریع، خودکارسازی فرآیندها و پایداری نیاز دارند