

# چرخه حیات تولید نرم افزار

## Software Development Life Cycle

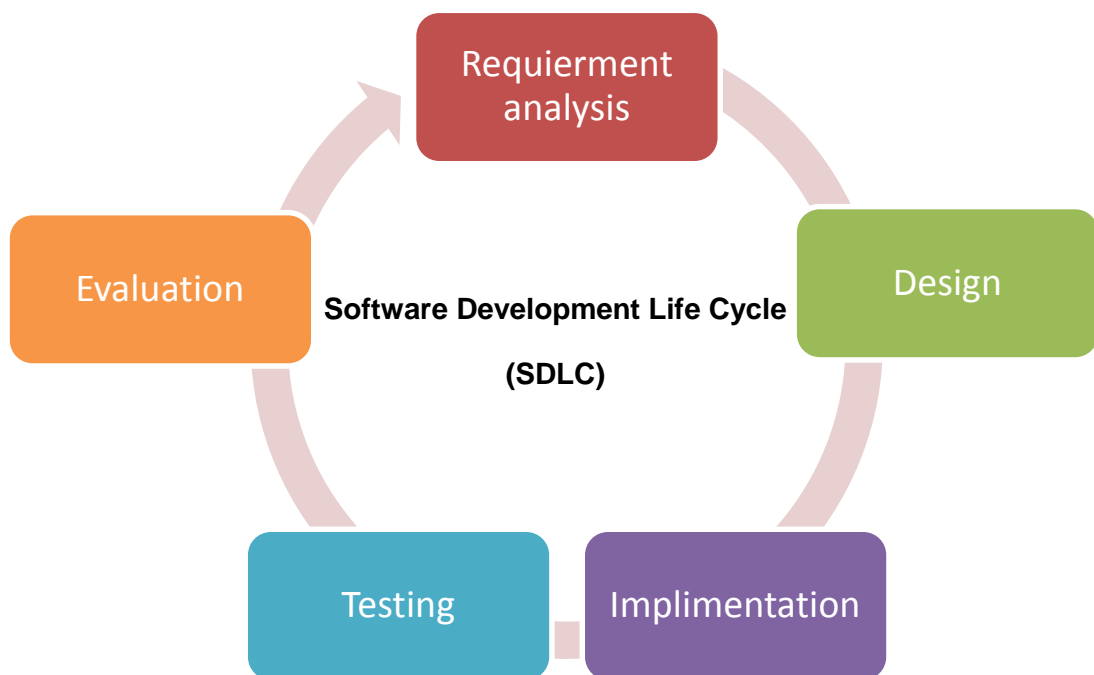


# چرخه حیات تولید نرم افزار

## Software Development Life Cycle

(SDLC)

### راهنمای آموزشی



ترجمه و گردآوری: سعید قیدرپور

## فهرست مطالب

صفحه	عنوان	ردیف
3	مروری کلی بر چرخه حیات تولید نرم افزار	1
4	چرخه حیات تولید نرم افزار چیست؟	2
6	مدلهای چرخه حیات تولید نرم افزار	3
7	<b>مدل آبشاری</b>	4
8	طرح مدل آبشاری	5
9	کاربرد مدل آبشاری	6
9	نظرات موافق و مخالف مدل آبشاری	7
11	<b>مدل تکراری یا افزایشی</b>	8
11	طرح مدل تکراری یا افزایشی	9
12	کاربرد مدل تکراری یا افزایشی	10
12	نظرات موافق و مخالف مدل تکراری یا افزایشی	11
14	<b>مدل حلزونی</b>	12
14	طرح مدل حلزونی	13
15	کاربرد مدل حلزونی	14
16	نظرات موافق و مخالف مدل حلزونی	15
17	<b>مدل V</b>	16
19	طرح مدل V	17
21	کاربرد مدل V	18
21	نظرات موافق و مخالف مدل V	19
23	<b>مدل بیگ بنگ</b>	20
23	طرح و کاربرد مدل بیگ بنگ	21
24	نظرات موافق و مخالف مدل بیگ بنگ	22
25	<b>مدل چابک</b>	23
27	مدل چابک در مقابل مدل‌های سنتی	24
28	نظرات موافق و مخالف مدل چابک	25
30	<b>مدل تولید سریع</b>	26
31	مدل تولید سریع چیست؟	27
31	طرح مدل تولید سریع	28
33	مدل سریع در مقابل مدل‌های سنتی	29
33	کاربرد مدل تولید سریع	30
34	نظرات موافق و مخالف مدل تولید سریع	31
35	<b>فشرده مطالب</b>	32
36	<b>کتاب مرجع</b>	33

## چرخه حیات تولید نرم افزار

### Software Development Life Cycle (SDLC)

چرخه حیات تولید نرم افزار روندی است که تحلیلگران فن آوری اطلاعات با استفاده از آن اقدام به تولید و طراحی سیستمهای نرم افزاری و انطباق آنها با نیازهای مشتریان نموده و در تحلیل خود نیازمندیهای دنیای واقعی را مد نظر قرار میدهند. تحلیلگران فن آوری اطلاعات در تمامی روند تولید یعنی آزمایش، تجزیه و تحلیل و نگهداری، کلیه جنبه های مثبت و منفی را ثبت و مورد توجه قرار میدهند.

SDLC حروف اختصاری Software Development Life Cycle میباشد که بمعنی چرخه حیات تولید نرم افزار یا سیستم میباشد و برخی از تحلیلگران آنرا روند تولید نرم افزار می نامند.

چرخه حیات نرم افزار چارچوبی است که کلیه عملیاتی را که در هر مرحله از تولید نرم افزار انجام میگردد تعریف مینماید. استاندارد 13307 یک استاندارد و معیار بین المللی جهت روند تولید نرم افزارها به شمار میرود.

هدف از تدوین این استاندارد تعریف کلیه وظایف مرتبط با چرخه حیات تولید نرم افزارها میباشد.

چرخه حیات تولید نرم افزار چیست؟

چرخه حیات تولید نرم افزار (SDLC) روندی است که در تولید و یا پروژه ویا تجدید طراحی یک نرم افزار در یک سازمان دنبال میگردد. این روند کلیه عملیات از جمله نحوه تولید، نگهداری، جایگزینی و یا تغییر و یا بهبود کیفی نرم افزار را در بر میگیرد.

چرخه حیات معرف شناسائی روش اصلاح و بهبود کیفی نرم افزار و روند تولید همه جانبه آن است.

تصویر صفحه بعد نشان دهنده مراحل مختلف چرخه حیات تولید نرم افزار میباشد.



## چرخه حیات تولید نرم افزار شامل مراحل ذیل می باشد.

### مرحله 1: برنامه ریزی و تجزیه و تحلیل نیازمندیها

تجزیه و تحلیل نیازمندیها، مهمترین و اساسی ترین مرحله در روند چرخه حیات تولید نرم افزار محسوب میگردد. این امر توسط کارشناسان ارشد و اعضای متبحر تیم با استناد به داده های ورودی مشتریان، بخش فروش و بررسی های بازار و متخصصین این حوزه از صنعت انجام میگردد. پس از این مرحله، اطلاعات گردآوری شده، جهت طراحی اساس و پایه پروژه و مطالعه امکان سنجی آن در حوزه های اقتصادی، عملیاتی و فنی مورد استفاده قرار میگردد.

طراحی تضمین کیفیت نرم افزار و همچنین شناسائی ریسکها و خطرات بالقوه پروژه نیز در مرحله برنامه ریزی انجام میگردد. نتیجه مطالعات و امکان سنجی ها موجب بدست آوردن دیدگاههای مختلف فنی خواهد بود که میتواند باعث پیاده سازی موفقیت آمیز پروژه با حداقل ریسک گردد.

### مرحله 2: تعریف نیازمندیها

زمانی که تجزیه و تحلیلها به پایان میرسد مرحله بعد یعنی تعریف دقیق نیازمندیها و مستند نمودن آن آغاز میشود. این مستندات شامل نیازهای نرم افزار مورد نظر بوده که مورد تائید مشتری و همچنین تحلیل گران بازار قرار گرفته است. سند اصلی در این مرحله سند مشخصات نیازمندیهای نرم افزار است که شامل کلیه نیازهای محصول بوده که در روند چرخه حیات پروژه می باید طراحی و فراهم گردد.

### مرحله 3: طراحی معماری نرم افزار

سند مشخصات نیازمندیهای نرم افزار (Software Requirement Specification (SRS یک سند مرجع برای معماری نرم افزار محسوب میگردد که در آن انواعی از بهترین معماریها جهت تولید نرم افزار مورد نظر تشریح گردیده است. کارشناسان معماری سیستم ها معمولاً براساس نیازمندیهای مندرج در مستند مذکور، یک و یا دو طرح پیشنهادی را جهت تعیین نهائی معماری مورد نیاز انتخاب و در مستند دیگری بنام سند مشخصات طرح Design Document Specification (DDS) ارائه و پیشنهاد مینمایند. سند مذکور توسط کلیه ذینفعان و براساس معیارهای گوناگون نظیر ارزیابی ریسک، پیمانیکی طرح (modularity) قابلیت استفاده در شرایط مختلف، محدودیت زمانی و بودجه مورد بررسی و ارزیابی قرار گرفته و سرانجام بهترین طراحی جهت تولید نرم افزار انتخاب میگردد.

در طراحی نرم افزار کلیه و تمامی جنبه های معماری و همچنین روشهای ارتباطی و جریان ارسال و دریافت داده ها از سامانه های دیگر (در صورت وجود آنها) مد نظر قرار میگردد. در معماری پیشنهادی، می باید طراحی داخلی کلیه مدل ها و پیمانه ها بنحوی کاملاً واضح و آشکار و با ذکر تمامی جزئیات در سند مشخصات طرح (DDS) درج گردد.

## مرحله 4 : تولید نرم افزار

در این مرحله از چرخه حیات تولید نرم افزار، روند عملیاتی آغاز و نرم افزار تولید میگردد. در این مرحله کد برنامه ریزی بهمان صورتی که در سند مشخصات طرح تاکید گردیده تهیه میگردد. چنانچه طرح بر مبنای جزئیات مندرج در سند مذکور و بصورت سازمان داده شده پیاده شود تولید نیز بدون هیچ مشکلی انجام خواهد شد تولید کنندگان نرم افزار میباید از رهنمودهای کد گزاری که بوسیله سازمان خود و ابزار برنامه نویسی تعریف شده است تبعیت نمایند. این تولیدکنندگان الزماً از ابزارهایی نظیر هم گردانها (compilers) مفسرها (interpreters) خطا زدها (debuggers) و غیره جهت تولید کدها استفاده خواهند نمود زبانهای مختلف برنامه نویسی سطح بالا نظیر C و ++C ، پاسکال، جاوا و PHP برای برنامه نویسی عموماً مورد استفاده قرار میگیرند. انتخاب زبان برنامه نویسی معمولاً با توجه به نوع نرم افزاری که میباید تولید شود انتخاب میگردد.

## مرحله 5 : آزمایش محصول

این مرحله معمولاً زیر مجموعه ای از کلیه مراحل دیگر در مدل‌های نوین چرخه حیات تولید نرم افزار محسوب میگردد. عملیات مربوط به آزمایش محصول اغلب در کلیه مراحل تولید محصول انجام میگردد. ولی آزمایش در این مرحله زمانی انجام میگردد که نقص یا کاستی هائی مشاهده و یا گزارش شده است. بدین لحاظ کارشناسان پس از پیگیری نقص و یافتن علل، نقص یا کاستی را بر طرف و مجدداً تا زمانی که محصول در سطح استاندارد است که در سند مشخصات نیازمندیهای نرم افزار مشخص گردیده است قرارگیرد نرم افزار را آزمایش مینمایند.

## مرحله 6 : پیاده سازی در بازار و نگهداری

زمانی که نرم افزار مراحل نهائی آزمایش را طی و آماده پیاده سازی شد، عرصه رسمی آن در یک بازار ویژه آغاز میگردد. گاهی اوقات پیاده سازی نرم افزار در یک سازمان بصورت مرحله ای انجام میگردد و این امر بستگی به استراتژی کسب و کار سازمان مذکور دارد. امکان دارد که یک نرم افزار در ابتدای امر در بخش محدودی از یک محیط تجاری واقعی پیاده سازی شود که این امر "آزمایش به شرط قبول کاربر" نامیده میشود. در این شرایط و با توجه به بازخوردها امکان دارد که پیشنهادات دیگری برای کارکرد بهتر محصول ارائه و تغییراتی متناسب با نیازهای بازار انجام گیرد. پس از پیاده سازی نرم افزار مرحله نگهداری و پشتیبانی از مشتری موجود آغاز میگردد.

چرخه حیات تولید نرم افزار دارای مدل‌های گوناگونی است و در روند تولید نرم افزار از یکی از مدل‌های طراحی و تعریف شده استفاده میگردد. عبارت مذکور گاهی بنام مدل‌های روند تولید نرم افزار نیز نامیده میشود. هر یک از مدلها دارای مراحل منحصر به فرد خود میباشد که موفقیت روند تولید نرم افزار را تضمین مینماید.

در ذیل به مهمترین و مشهور ترین مدل‌های چرخه حیات تولید نرم افزار در این حوزه صنعت اشاره میشود:

- مدل آبشاری waterfall model
- مدل تکراری iterative model
- مدل حلزونی spiral
- مدل V-model V
- مدل بیگ بنگ Big Bang model

سایر روشهای و یا مدل‌های چرخه حیات تولید یا تولید نرم افزار عبارتند از :

- مدل سریع Agile model
- مدل تولید سریع برنامه کاربردی Rapid Application Development (RAD)

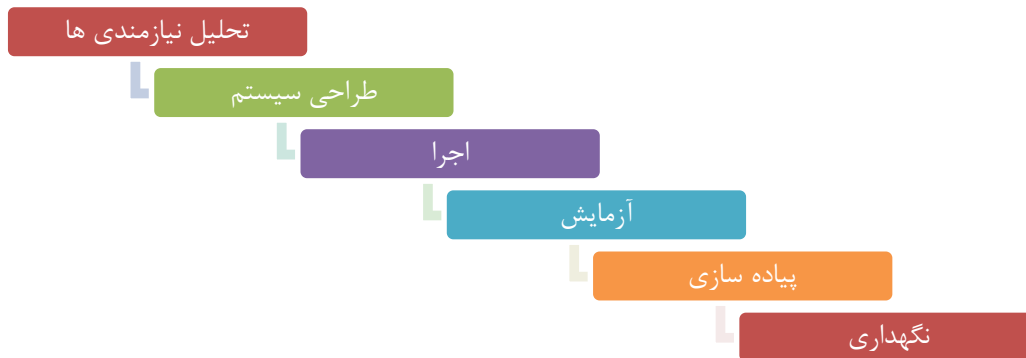
### 1: مدل آبشاری چرخه حیات تولید نرم افزار

مدل آبشاری اولین مدلی بود که در این زمینه ارائه گردید. این مدل به نام چرخه خطی و پی در پی نیز شهرت دارد فراگیری و استفاده از این مدل بسیار آسان و ساده می باشد. در مدل آبشاری، هر مرحله می باید قبل از اینکه مرحله بعدی شروع شود پایان یابد و بین هیچ یک از مراحل تداخل یا هم پوشانی وجود ندارد. مدل آبشاری اولین روش در زمینه چرخه حیات تولید نرم افزار محسوب شده و نرم افزارهای اولیه با این روش تهیه می شدند. مدل آبشاری نشانگر روند تهیه و تولید نرم افزار به شیوه خطی و پی در پی است و بدین لحاظ هر مرحله در این مدل زمانی آغاز می گردد که مرحله قبلی تکمیل شده باشد. بر این اساس تمامی مراحل مستقل از یکدیگر بوده و هیچ مرحله ای با مرحله دیگر هم پوشانی ندارد.

### طرح مدل آبشاری

روش آبشاری اولین مدل چرخه حیات تولید نرم افزار است که جهت تضمین موفقیت پروژه، در مهندسی نرم افزار مورد استفاده قرار گرفت. در این مدل روند تولید نرم افزار به مراحل جدا از یکدیگر تقسیم می گردند. دیاگرام ذیل نشاندهنده مراحل مختلف مدل آبشاری می باشد.





### مراحل پی در پی در مدل آبخاری عبارتند از :

- جمع آوری و تجزیه و تحلیل نیازمندیها: کلیه نیازمندیهای لازم جهت تولید نرم افزار در این مرحله جمع آوری و در سند مشخصات نیازمندیها ثبت میگردد.
- طراحی سیستم : مشخصات کامل نیازمندیها که در مرحله اول جمع آوری شده است در این مرحله مورد مطالعه دقیق قرار گرفته و طراحی سیستم انجام و طرح لازم بدست میآید. در طراحی سیستم نیازهای سخت افزاری و نرم افزاری را مشخص شده و باعث میگردد که معماری کلی سیستم تعریف گردد.
- اجرا : در این مرحله با استفاده از وروده‌های طرح سیستم، ابتدا برنامه هائی به مقیاس کوچک بنام واحدها نوشته میشود که در مرحله بعدی با یکدیگر ادغام میگرددند. بعد از ادغام بمنظور بررسی عملکرد مورد آزمایش قرار میگیرند. این مرحله را مرحله آزمایش واحد ها نیز مینامند.
- ادغام و آزمایش : کلیه واحد هائی که در مرحله تولید و اجرا با یکدیگر ادغام شده و به شکل یک سیستم در آمده اند مورد آزمایش قرار میگیرند. پس از ادغام، کل سیستم جهت رفع نقایص و کمبود ها مجدداً آزمایش میگردد.
- پیاده سازی : زمانی که کلیه آزمایشات با موفقیت به انجام رسید، محصول در محیطی که مشتری پیش بینی نمود پیاده سازی میگردد و یا اینکه مستقیماً به بازار فروش ارسال میگردد.

- نگهداری : برخی از مشکلات وجود دارند که صرفاً در محیط مشتری بروز مینمایند. برای رفع این مشکلات اغلب تغییرات و اصلاحاتی (patches) انجام میگردد و گاهی برای ارتقای سیستم نسخه جدیدی تهیه میگردد. تغییرات و یا نسخه جدید مجدد در محیط کار مشتری نصب و راه اندازی میگردد.
- کلیه مراحل فوق آنچنان پی در پی انجام میگردد که در کل مثل جریان یک آبشار از یک مرحله به مرحله دیگر عبور و در کل بصورت یک امر مداوم از ابتدا تا انتها حرکت مینماید. در این مدل مرحله بعد زمانی آغاز میشود که مرحله قبل هدفهای مرحله بعد و نحوه رسیدن به آن را مشخص نموده و مورد تأیید قرار گرفته باشد. بنابراین مرحله های مدل آبشاری با یکدیگر اصطکاک ندارند.

### کاربرد مدل آبشاری

هر نرم افزاری دارای چرخه حیات مطلوب و متفاوت خود میباشد و این امر مبتنی بر یک سری عوامل داخلی و خارجی میباشد. برخی شرایط که استفاده از مدل آبشاری را امکان پذیر و متناسب میسازد عبارتند از:

- نیازها کاملاً روشن و ثابت و مستند به عوامل شناخته شده میباشدند.
- تعریف محصول ثابت و پایدار است
- فن آوری کاملاً درک شده و تغییر پذیر نمیباشد
- نیازها بصورت مبهم و غیر واضح نمیباشند
- منابع و کارشناسان مجرب فراوان جهت پشتیبانی از محصول وجود دارند
- طول عمر پروژه کوتاه است

### موافقان و مخالفان مدل آبشاری

مزیت :

مزیت تولید نرم افزار بر طبق مدل آبشاری این است که در این مدل عملیات هر یک جداگانه انجام و کنترل میگرددند. در این مدل میتوان یک برنامه کاری تنظیم و برای هر مرحله نقطه آغاز و پایانی در نظر گرفت و محصول میتواند براساس برنامه مذکور مرحله به مرحله پیشرفت و هر یک از مراحل بعد از کنترل مرحله بعد را بدنبال داشته باشد.

عدم مزیت :

عدم مزیت مدل آبخاری عدم امکان بازگشت و تجدید نظر در تولید نرم افزار یا محصول است. هنگامیکه یکی از مراحل در جریان تست و آزمایش قرار دارد، بازگشت به عقب و اعمال تغییر در فعالیتهایی که بصورت مستند در نیامده است بسیار مشکل میباشد.

در جدول زیر فهرستی از نظر موافقین و مخالفین مدل آبخاری مشاهده میگردد :

نظرات موافق	نظرات مخالف
<ul style="list-style-type: none"> <li>• این مدل بسیار ساده و درک و استفاده از آن نیز آسان است. مدیریت مدل بلحاظ استحکام مدل آسان است چون هر یک از مراحل دارای ارقام خاص قابل تحویل و روند بررسی خاص خود میباشد.</li> <li>• مراحل و روند ها در زمان خود پردازش و تکمیل میگردند.</li> <li>• اجرای پروژه های کوچکتر در صورتیکه نیازها بخوبی فهمیده شوند بسیار آسان است.</li> <li>• مراحل را میتوان بخوبی و روشنی از یکدیگر تفکیک نمود.</li> <li>• مراحل اساسی بخوبی قابل درک میباشند.</li> <li>• ترکیب و تعریف وظایف بسیار آسان است.</li> <li>• روندها و نتایج را میتوان بخوبی مستند نمود.</li> </ul>	<ul style="list-style-type: none"> <li>• هیچ نرم افزاری در این مدل تا پایان چرخه حیات تولید نمیگردد.</li> <li>• ریسک و عدم اطمینان در این مدل بسیار بسیار بالا است.</li> <li>• مدل خوبی برای پروژه های پیچیده و یا شی گرا میباشد .</li> <li>• برای پروژه های طولانی مدت و مداوم مدل مناسبی نیست.</li> <li>• برای پروژه هایی که نیازمندیها و تامین آنها در معرض ریسک بالا و یا متوسط هستند و نیازمندیها باید تغییر یابند مدل مناسبی نمیشود. پس بنابراین ریسک و عدم اطمینان در این مدل بسیار بالا میباشد.</li> <li>• اندازه گیری پیشرفت کار در هر مرحله مشکل است.</li> <li>• این مدل نمیتواند سازگار با تغییر نیازها باشد.</li> <li>• تعدیل هدف در طول چرخه حیات میتواند به انجام پروژه پایان دهد.</li> </ul>

## 2: مدل تکراری چرخه حیات تولید نرم افزار (Iterative model)

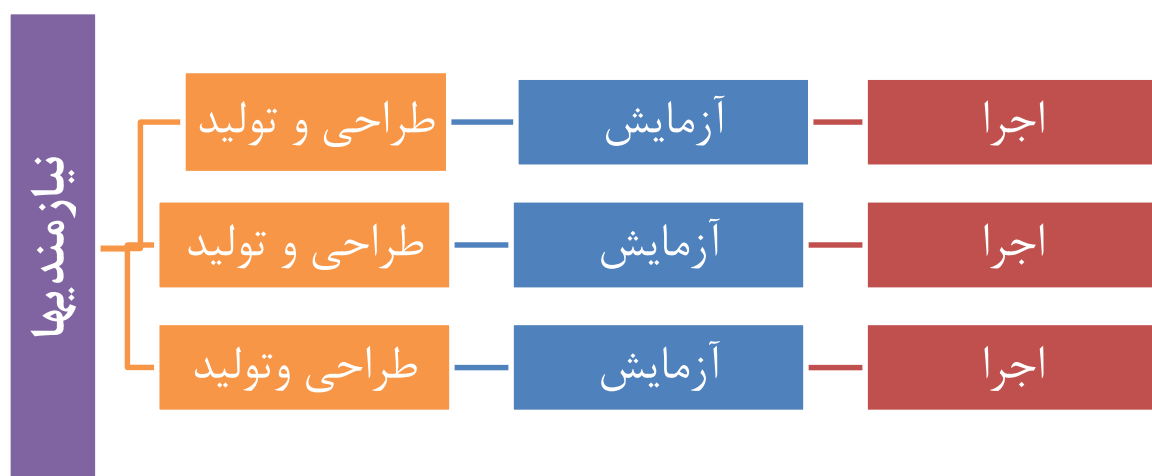
در مدل تکراری، روند تکرار از طریق اجرای یک مجموعه کوچک نرم افزار آغاز و با افزایش نیازها نسخه های متکامل تر افزوده میشود تا اینکه سیستم بصورت کامل اجرا و جهت پیاده سازی آماده گردد.

در مدل چرخه حیات تکراری، تلاش برای شروع با تمامی نیازمندیها انجام نمیگردد، بلکه تولید با شناسایی و اجرای صرفاً بخشی از نرم افزار شروع و سپس بررسی برای شناخت نیازهای بیشتر آغاز میگردد. این روند سپس تکرار شده و در پایان هر مرحله تکرار مدل، نسخه جدیدی از نرم افزار تهیه میشود.

### طرح مدل تکراری

در این مدل یعنی در روند تکراری تولید نرم افزار عملیات با اجرای ساده یک مجموعه از نیازهای نرم افزاری شروع و رفته رفته نسخه های متکامل تری تهیه و اضافه میشود تا نهایتاً سیستم بطور کامل پیاده سازی شود. در هر مرحله از تکرار، تغییراتی در طرح اعمال و توانائی های عملیاتی جدیدی اضافه میشود. ایده اصلی در این روش، تولید سیستم از طریق چرخه های تکرار در مدت زمانی کمتر میباشد.

در تصویر صفحه بعد نمونه ای از مدل تکراری چرخه حیات تولید نرم افزار مشاهده میگردد.



روند تولید افزایشی یا تکراری ترکیبی از طراحی تکراری جهت ساختن مدل تولید میباشد. در این مدل هنگام تولید نرم افزار چند چرخه تولید نرم افزار بصورت هم زمان انجام میگردد و این روند را میتوان بعنوان مدل تکاملی و یا روش صعودی نیز نام گذاری کرد.

در مدل تکراری، کلیه نیازمندیهای را در بخشهای مختلف تقسیم میکنند. طی هر دوره تکرار یا تولید نرم افزار از مرحله نیازمندیها به بخش های طراحی، اجرا و تست و آزمایش عبور مینماید و نسخه جدید نرم افزار وظایف و عملیات جدیدی را نسبت به نسخه قبل اضافه مینماید. این روند تا تکمیل شدن کامل و انطباق با نیازمندیها ادامه مییابد. کلید کاربرد موفقیت آمیز چرخه حیات تولید نرم افزار شناخت کامل و صحیح نیازمندیها تا تأیید و تصدیق آنها و تست و آزمون هر نسخه از نرم افزار با نیازمندیها در هر چرخه از مدل میباشد. در هر مرحله که نرم افزار در چرخه های پی در پی پیش میرود آزمایشها نیز جهت تأیید هر نسخه از نرم افزار نیز میباید تکرار شوند.

### **کاربرد مدل تکراری**

مدل تکراری و یا افزایشی نظیر سایر مدل‌های موجود در چرخه حیات تولید نرم افزار دارای کاربردهای ویژه ای میباشد. این مدل در اغلب اوقات در سناریوهای ذیل مورد استفاده قرار میگیرد.

- نیازمندیهای کل سیستم تعریف و شناسائی شده باشند.
- نیازمندیهای عمده می باید تعریف شده و عملکردهای جدید و یا ارتقا جدید در نرم افزار در طی زمان اضافه شوند.
- محدودیت های زمانی در بازار باید مد نظر قرار گیرد.
- فن آوریهای جدید می باید توسط تیم تولید در زمان کار بر روی پروژه مورد استفاده قرار گیرد.
- مجموعه ای از منابع انسانی و مهارت‌های مورد نیاز که در وضعیت موجود وجود ندارند در هر مرحله از حیات از طریق انعقاد قرارداد در انجام پروژه به کار گرفته شوند.
- برخی ویژگی ها و یا اهداف پر ریسک وجود دارند که در آینده می باید نسبت به تغییر آنها اقدام نمود.

### **موافقان و مخالفان مدل تکراری**

امتیاز این مدل در این است که در اولین مرحله تولید، مدلی از کارکرد سیستم وجود دارد که میتوان از روی آن کاستی های عملیاتی و یا طراحی را یافت. یافتن این موارد در مراحل اولیه تولید، کارشناسان را قادر میسازد با یک بودجه محدود اقدامات اصلاحی انجام دهند.

عدم امتیاز این مدل از چرخه حیات تولید و یا تولید نرم افزار امنیت که این مدل صرفاً برای پروژه های عظیم و حجیم کاربرد دارد. علت این مسئله اینست که شکستن یک سیستم نرم افزار ی کوچک و تقسیم آن به مدل های سرویس دهنده کوچکتر بسیار مشکل میباشد.

جدول زیر نشان دهنده موافقین و مخالفین مدل تکراری چرخه حیات تولید نرم افزار است.

نظرات موافق	نظرات مخالف
<ul style="list-style-type: none"> <li>• برخی عملیات را میتوان سریعاً و در مراحل اولیه چرخه حیات تولید داد.</li> <li>• نتایج عملیات بسیار سریع و بصورت دوره ای بدست میآید.</li> <li>• در این مدل میتوان تولید را بصورت موازی برنامه ریزی نمود.</li> <li>• پیشرفت تولید را میتوان اندازه گیری کرد.</li> <li>• تغییر طرح نهائی و نیازها با هزینه کمتری انجام میگردد.</li> <li>• آزمایش و اشکال زدائی در یک زمان کوتاهتر انجام میگردد.</li> <li>• ریسکها در یک دوره کوتاهتر شناسائی و حل میگرددند و مرحله تکراری به سادگی مدیریت میشود.</li> <li>• مدیریت ریسک در این مدل آسانتر است و مدیریت ریسکهای بزرگتر قبل از دیگر ریسکها انجام میگردد.</li> <li>• در این مدل در هر مرحله بخشی از محصول تهیه میگردد.</li> <li>• حل مسائل، چالشها و ریسکها که در هر مرحله شناسائی میگرددند میتوانند در مرحله بعدی مورد استفاده قرار گیرد.</li> <li>• تجزیه و تحلیل ریسکها بنحو مطلوبتر انجام میگردد.</li> <li>• این مدل از تغییر نیازمندیها پشتیبانی مینماید.</li> <li>• زمان انجام عملیات ابتدائی کوتاهتر است.</li> <li>• کاربرد این مدل برای پروژه های بلند مدت و یا حیاتی مناسب تر است.</li> <li>• در خلال چرخه حیات، نرم افزار سریعتر تولید شده و ارزیابی مشتری را آسانتر میسازد.</li> </ul>	<ul style="list-style-type: none"> <li>• در این مدل منابع بیشتری مورد نیاز میباشد.</li> <li>• گرچه هزینه اعمال تغییر در این مدل کمتر است ولی جهت تغییر نیازمندیها چندان مطلوب نمیباشد.</li> <li>• این مدل نیاز به توجه بیشتری از سوی مدیریت دارد.</li> <li>• در معماری و یا طراحی سیستم ممکن است مسائل بیشتری بوجود آید زیرا در شروع تمامی چرخه های حیات نمیتوان تمامی نیازمندیها را گردآوری نمود.</li> <li>• تعریف هر یک از مراحل تکراری نیاز به تعریف سیستم بطور کلی دارد.</li> <li>• این مدل برای پروژه های کوچک مناسب نیست.</li> <li>• پیچیدگی عملیات مدیریتی بیش از سایر مدلها است.</li> <li>• پایان پروژه قابل پیش بینی نیست و این امر دارای ریسک میباشد. منابع انسانی کاملاً متبخر جهت تجزیه و تحلیل ریسک مورد نیاز میباشد.</li> <li>• پیشرفت پروژه کاملاً وابسته به مرحله تجزیه و تحلیل ریسک میباشد.</li> </ul>

### 3: مدل حلزونی چرخه حیات تولید نرم افزار

مدل حلزونی را میتوان ترکیبی از مدل چرخه حیات تکراری و جنبه های کنترل شده و اصولی مدل آبشاری دانست. مدل حلزونی ترکیبی از مدل روند تولید تکراری و مدل تولید خطی و متوالی یعنی مدل آبشاری با تاکید بسیار بر روی تجزیه و تحلیل ریسک میباشد.

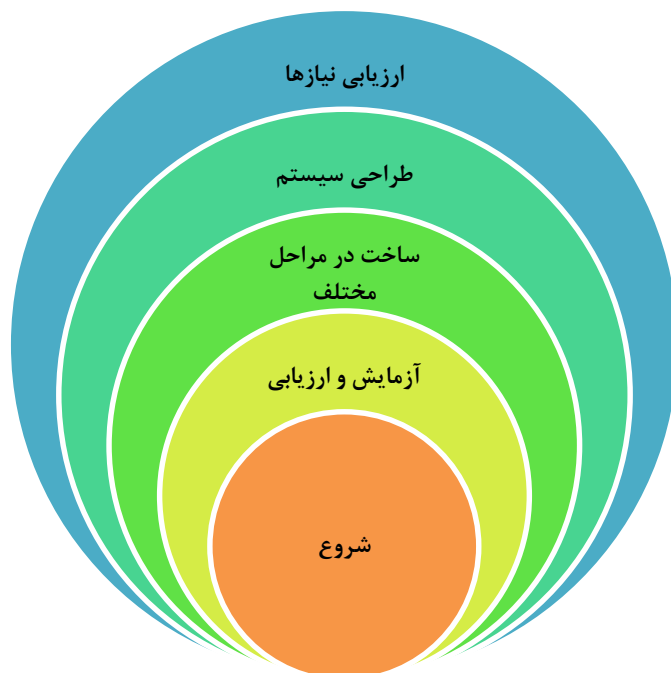
در این مدل هر بار تکرار عملیات در حلقه حلزونی میتوان محصول یا نرم افزار را ارتقا داده و در هر بار آنرا پالایش داد.

#### طرح مدل حلزونی

مدل حلزونی دارای چهار مرحله میباشد. هر پروژه تولید نرم افزار بصورت مکرر از این مراحل عبور کرده و مسیر حلزونی را طی مینماید. این مراحل عبارتند از:

- شناسائی : این مرحله با گردآوری نیازمندیهای کسب و کار شروع میشود. در مسیر حلزونی بعدی زمانی که محصول کاملتر گردید، شناسائی نیازمندیهای سیستم، سیستم های فرعی و همچنین نیازمندی واحدها در این مرحله انجام میگردد. درک نیازمندیهای سیستم از طریق ارتباط دائم مشتری و تحلیلگر سیستم امکان پذیر میباشد. در پایان این مسیر محصول یا نرم افزار در بازار مورد نظر پیاده سازی میشود.
- طرح : این مرحله با طراحی ذهنی و در مرحله اولیه انجام میگردد که شامل طراحی معماری، طراحی منطقی واحدها و بخشهای مختلف محصول (پیمانه ها) طراحی فیزیکی و تهیه طرح نهائی است که در مراحل بعدی انجام میگردد.
- ساخت یا تولید : در این مرحله، نرم افزار یا محصول واقعی با توجه به طرح نهائی و با در نظر گرفتن نیازهای مشتری تولید میگردد. پس از تولید نرم افزار یا محصول جهت آزمایش با درج شماره نسخه جهت دریافت نظر مشتری و اعمال تغییرات نهائی ارسال میگردد.
- بررسی و ارزیابی و تجزیه و تحلیل ریسک : تجزیه و تحلیل ریسک عبارتست از ، شناسائی، پایش امکانات فنی و مدیریت ریسک از جمله انطباق محصول با طرح نهائی و محاسبه هزینه با برآورد اولیه . در این مرحله مشتری نرم افزار را ارزیابی و نظرات نهائی خود را اعلام مینماید.

تصویر ذیل نشاندهنده مدل حلزونی چرخه حیات تولید نرم فزار است.



### کاربرد مدل حلزونی

مدل حلزونی بنحو بسیار گسترده ای در صنعت نرم افزار مورد استفاده میباشد زیرا این مدل همگام با روند تولید طبیعی هر محصول بوده و در حالیکه کمترین ریسک را برای مشتری میتواند به دنبال داشته باشد، منافع شرکت تولید کننده را نیز به همراه دارد. ذیلاً به چند نمونه از کاربرد این مدل اشاره میگردد:

- در مواردیکه بودجه دارای محدودیت است و هزینه ها جزو مسائل حیاتی بشماره میرود و ارزیابی نیز با اهمیت تلقی میگردد این مدل میتواند مطلوب باشد.
- این مدل برای پروژه پر ریسک و نیز با ریسک متوسط، مدلی موثر محسوب میگردد.
- این مدل برای پروژه های بلند مدت نیز مدلی مطلوب بشمار میرود. زیرا تعهدات انجام این پروژه بلحاظ ایجاد تغییرات بالقوه در الویت های اقتصادی بسیار زیاد بوده و نیازمندیها نیز در طول زمان دچار تغییر میگردد.
- در مواردیکه مشتری به نیازمندیهای خود کاملاً اطمینان ندارد نیز این مدل میتواند کاربرد داشته باشد.
- در مواردیکه نیازمندیهای مشتری پیچیده و روشن ساختن آنها نیاز به ارزیابی بیشتر دارد نیز این مدل میتواند کاربرد داشته باشد.
- در مورد خطوط تولید محصول جدید که میباید در مراحل مختلف انجام گرفته و در هر مرحله نظر مشتری کسب شود نیز این مدل میتواند مورد استفاده قرار گیرد.



- در مواردیکه در چرخه تولید محصول انتظار ایجاد تغییرات اساسی می‌رود نیز این مدل موثر باشد.

### موافقین و مخالفین مدل حلزونی

یکی از مهمترین فواید مدل چرخه حیات حلزونی اینست که در این مدل میتوان عناصر تولید را در هر زمان که مورد شناسایی و یا در دسترس قرار گیرد به محصول اضافه نمود و این امر تضمین مینماید که بین نیازمندیهای قبلی و طرح تدوین شده مغایرتی به وجود نخواهد آمد.

یکی دیگر از جنبه های مثبت این مدل اینست که کاربران را در همان اوایل کار درگیر کار بر روی تولید سیستم مینماید. از سوی دیگر، این مدل، مدیریت را در تکمیل پروژه بسیار سخت مشغول نموده و این خطر وجود دارد که مدل حلزونی مدیریت را در یک جنبه نا معین گرفتار نماید، بدین خاطر این مدل نیاز به نظمی سخت در ایجاد تغییرات و دامنه درخواست تغییرات دارد و این نظم نقش بسیار مهمی در تولید و پیاده سازی موفقیت آمیز پروژه بعهد خواهد داشت تصویر ذیل نشاندهنده نظرات موافق و مخالف این مدل میباشد.

نظرات موافق	نظرات مخالف
<ul style="list-style-type: none"> <li>• تغییر نیازمندیها را میتوان با طرح وفق داد.</li> <li>• استفاده گسترده از پیش الگوها امکان پذیر است.</li> <li>• مقررات را میتوان بطور صحیح اعمال نمود.</li> <li>• کاربران بسیار زود سیستم را مشاهده خواهند نمود.</li> <li>• تولید را میتوان به بخشهای کوچکتر تقسیم و یا بخشهای پر ریسک را زودتر انجام داد که این امر مدیریت ریسک را آسانتر میسازد.</li> </ul>	<ul style="list-style-type: none"> <li>• مدیریت در این مدل بسیار پیچیده میباشد.</li> <li>• پایان پروژه را نمیتوان بزودی تخمین زد.</li> <li>• با پروژه های کوچک و یا طرحهای کم ریسک سازگار نبوده و هزینه آن برای پروژه های کوچک بسیار بالا میباشد.</li> <li>• روند عملیات پیچیده میباشد.</li> <li>• تعداد بسیاری از مراحل میانی نیازمند مستند سازی بیش از اندازه میباشد.</li> </ul>

#### 4: حلقه حیات تولید نرم افزار - مدل - V

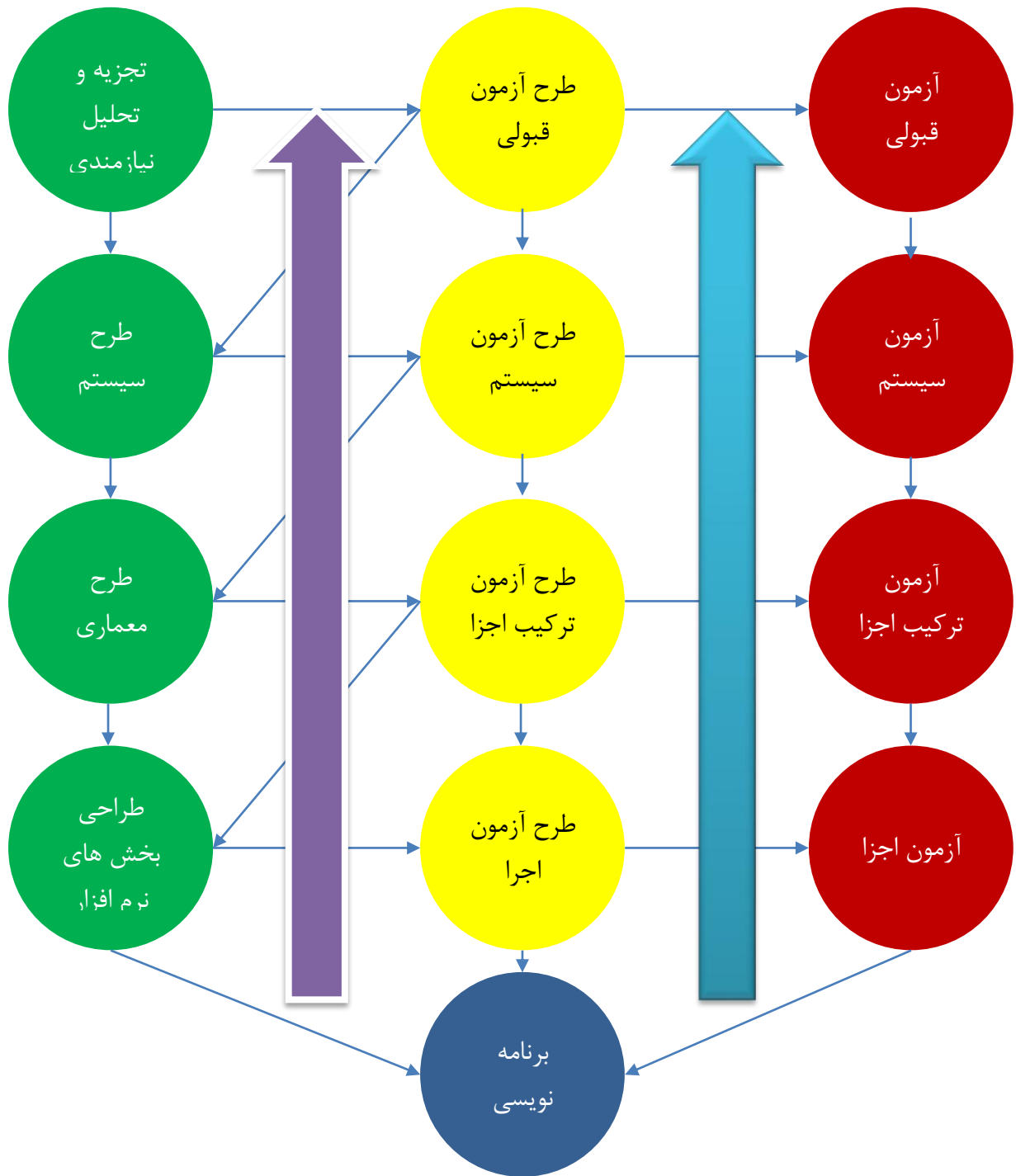
مدل V در چرخه حیات تولید نرم افزار مدلی است که انجام روند عملیات به شیوه پی در پی و شکل V صورت می پذیرد. این شیوه به مدل شناسائی و تصدیق اعتبار نیز شهرت دارد.

مدل V را میتوان به نوعی مدل بسط و تعمیم مدل آبشاری محسوب نمود. این مدل مبتنی بر تلفیق و تجمیع مرحله آزمون جهت هر یک از مراحل تولید میباشد. بدین معنی که برای هر یک از مراحل چرخه حیات یک مرحله آزمون مربوط به همان مرحله تولید وجود دارد. این مدل، روشی منظم و منضبط است و مرحله بعد صرفاً زمانی آغاز میشود که مرحله قبلی بطور کامل تکمیل شده باشد.

#### طرح مدل V

در مدل V امر آزمون جهت هر مرحله از چرخه حیات بصورت موازی برنامه ریزی می شود بنابراین مرحله شناسائی در یک طرف V و مرحله تصدیق اعتبار در سوی دیگر V قرار دارند. مرحله برنامه نویسی این دو مرحله در دو طرف مدل V را به یکدیگر مرتبط میسازد

تصویر زیر نشاندهنده مراحل مختلف چرخه حیات تولید نرم افزار در مدل V میباشد.



### تجزیه و تحلیل نیازهای کاری:

تجزیه و تحلیل نیازها: اولین مرحله از چرخه تولید نرم افزار است که در آن نیازهای مشتری و انتظارات او از محصول بدست میآید. این مرحله در برگیرنده ارتباط مداوم با مشتری جهت درک کامل جزئیات انتظارات و نیازهای عینی و کامل وی میباشد. این امر از اهمیت بسیاری برخوردار است و نیاز به مدیریت دقیق و حساب شده دارد، زیرا اغلب مشتریان از نیازها و انتظارات خود بطور کاملاً دقیق مطلع نبوده و یا اطمینان کامل ندارد برنامه ریزی و طراحی و آزمون قبولی در این مرحله انجام میگیرد زیرا نیازهای کاری را میتوان بمنزله ورودی آزمون قبولی مورد استفاده قرار داد.

طرح سیستم: زمانی که نیازمندیها بنحو روشن و مشروح و دقیق مشخص گردید، آنگاه میتوان طراحی کل سیستم را آغاز کرد. طراحی سیستم شامل درک و تشریح کامل سخت افزار و ابزار ارتباطی مورد نیاز محصول در مرحله تولید میباشد. طرح آزمون سیستم نیز بر مبنای طراحی سیستم تدوین میگردد. انجام این امور در مراحل اولیه زمان کافی را جهت اجرای آزمایش در مراحل بعد باقی میگذارد.

طرح معماری: مشخصات معماری و ویژگی های نرم افزار در این مرحله طراحی میشود. معمولاً در این مرحله بیش از یک طرح فنی تدوین و پیشنهاد میگردد ولی در نهایت با در نظر گرفتن امکانات فنی و مالی تصمیم نهائی اتخاذ میگردد. طرح سیستم سپس به واحد ها و پیمانان های کوچکتر تقسیم میگردد که هر یک دارای قابلیت های متفاوت میباشد. این امر طراحی در سطوح بالا نیز مینامند (HLD).

انتقال داده ها و ارتباط بین پیمانان ها و واحدهای داخلی سیستم با جهان خارج (سایر سیستمها) در این مرحله به روشنی مشخص میگردد. آزمون ادغام و یکپارچگی سیستم میتواند در این مرحله طراحی و مستند گردد.

طرح واحد ها و پیمانان ها: در این مرحله طرح داخلی کلیه واحد ها و پیمانان های سیستم با جزئیات کامل مشخص و تدوین میگردد. این امر را طراحی در سطوح پائین نیز مینامند (LLD). در این طراحی آنچه که از اهمیت فوق العاده برخوردار است، این است طرح هر پیمانان باید در معماری سیستم با سایر پیمانان ها یا واحد ها و همچنین با سایر سیستمهای خارجی سازگاری داشته باشد. آزمون واحدها بخشی اساسی در روند تولید محسوب و باعث حذف اکثر نقایص و خطاها در همان مراحل اولیه میگردد. آزمون واحد ها میتواند در این مرحله و بر مبنای طراحی واحد های داخلی تدوین گردد.

## مرحله برنامه نویسی

برنامه نویسی و یا کد نویسی عملی واحد های سیستم که در مرحله طرح، تهیه گردیده در مرحله برنامه نویسی انجام میگردد. بهترین و مطلوب ترن زبان برنامه نویسی بر مبنای نیازمندیهای سیستم و معماری آن انتخاب میگردد. برنامه نویسی بر اساس استانداردها و راهنمای عملیات کد نویسی انجام میگردد.

کدها قبل از اینکه بصورت نرم افزار در انباره مستندات قرار گیرند جهت بهینه سازی و کسب بهترین کارایی برای دفعات متعدد و بی شمار مورد بررسی و تجدید قرار میگیرند.

## مراحل تأیید اعتبار

ذیلاً به مراحل مختلف تأیید و تصدیق اعتبار در مدل V اشاره میشود:

- آزمون واحد : آزمایشهای واحد که در مرحله طراحی پیمانه ها تدوین شده در خلال مرحله تأیید اعتبار اجرا میگردند. تست واحد آزمایش در سطح کد میباشد که میتواند در حذف اشکال و خطاها در مراحل اولیه یاری رساند گرچه با این تست، کشف کلیه عیوب و نقایص ناممکن میباشد.
- آزمون یکپارچگی : آزمون یکپارچگی با مرحله طراحی معماری نرم افزار مرتبط میباشد. آزمون یکپارچگی اصولاً جهت حصول اطمینان از هم زیستی و ارتباط پیمانه های داخلی در درون سیستم انجام می پذیرد.
- آزمون سیستم : آزمون سیستم مستقیماً با مرحله طراحی سیستم مرتبط میباشد. آزمون سیستم عملکرد سیستم و ارتباط آن با سیستمهای خارجی را کنترل مینماید. در بسیاری از موارد سازگاری سخت افزار و نرم افزار در هنگام اجرای آزمون سیستم کشف و آشکار میشوند.
- آزمون قبولی : آزمون قبولی با مرحله تجزیه و تحلیل نیازمندیهای کاری مرتبط بوده و شامل آزمون محصول در محیط کاری کاربر میگردد. آزمون قبولی، سازگاری محصول با سایر سیستمهای موجود در محیط کاربر را آشکار میسازد. این امر مسائل غیر عملیاتی نظیر کاستی و نقص بارگذاری یا قراردادن اطلاعات در منبع ذخیره و یا نقص در کارایی را در محیط واقعی کاربر روشن میسازد.

## کاربرد مدل V

کاربرد مدل V تقریباً مشابه مدل آبشاری میباشد، زیرا هر دو مدل از نوع متوالی و ترتیبی میباشد. نیازمندیها نیز قبل از اینکه پروژه شروع شود روشن و آشکار میباشند زیرا در هر دو مدل معمولاً برگشت به قبل و اعمال تغییرات بسیار پر هزینه میباشد. از این مدل معمولاً در حوزه تولید پزشکی و تولیدات داروئی استفاده میشود زیرا حوزه های مذکور حوزه هائی موكداً منضبط میباشند. سناریوهای زیر جهت استفاده از مدل V بسیار مناسب میباشد.

- نیازمندیها بنحوی تعریف شده و ثابت و بصورت مستند درآمده باشند.
- تعریف محصول ثابت و پایدار باشد.
- فن آوری بصورت پویا نبوده و تسط تیم پروژه بنحوی اثبات شده باشد.
- هیچ مقررات دو پهلو، مبهم یا تعریف نشده ای وجود نداشته باشد.
- پروژه کوتاه مدت باشد.

## موافقین و مخالفین مدل V

یکی از امتیازات مدل V این است که درک و کاربرد آن آسان میباشد. سادگی این مدل مدیریت آنرا نیز سهل تر میسازد. عدم امتیاز آن غیر قابل انعطاف بودن در مقابل تغییرات است و بهمین دلیل در دنیائی که نیازمندیها دائم در حال تغییر بوده و این امر در جهان پویای امروز متدوال است انجام تغییرات در این مدل عملی پر هزینه محسوب میگردد.

جدول زیر نشاندهنده نظرات موافق و مخالف مدل V میباشد.

نظرات موافق	
<ul style="list-style-type: none"><li>• این مدل دارای نظم و ترتیب کاملی است و مراحل در زمان تعیین شده تکمیل میگردد.</li><li>• برای پروژه های کوچکتر مناسب است زیرا در این پروژه ها نیازمندیها بخوبی میتواند تعریف شوند.</li><li>• درک و کاربرد آن سهل و آسان است. مدیریت این مدل بدلیل استحکام و قدرت ثبات آسان است. هر مرحله در این مدل دارای روندی مشخص جهت بررسی و تحویل میباشد.</li></ul>	<ul style="list-style-type: none"><li>• ریسک بالا و عدم انعطاف در قبال تغییرات.</li><li>• مدل نامناسب برای پروژه های پیچیده و موجودیت گرا (object oriented).</li><li>• مدلی ضعیف جهت پروژه های طولانی و مداوم.</li><li>• مدلی غیر مناسب برای پروژه هائی که تغییر نیازمندیهای آنها توام با ریسک بالا میباشدند.</li><li>• برگشت و تغییر کاربرد مدل که در هر مرحله آزمون قرارداد با اشکال فراوان روبرو میشود.</li><li>• تا آخرین مرحله چرخه حیات هیچ نرم افزاری تولید نمی گردد.</li></ul>



## 5: چرخه حیات تولید نرم افزار - مدل بیگ بنگ Big Bang

مدل بیگ بنگ در چرخه حیات تولید نرم افزار هیچ روند خاصی را دنبال نمیکند. تولید نرم افزار هنگامی آغاز میشود که وجه آن دریافت و عملیات بعنوان ورودی انجام و خروجی آن ارائه نرم افزاری است که ممکن است با نیازهای مشتری انطباق نداشته باشد.

مدل بیگ بنگ در چرخه حیات تولید نرم افزار، مدلی است که نیاز به برنامه ریزی اندکی داشته و پس از ارائه نرم افزار هیچگونه تولید بعدی انجام نمی گیرد. حتی مشتری دقیقاً به آن چه نیاز دارد اطمینان نداشته و نیازها بدون هیچگونه تجزیه و تحلیلی صرفاً در تخیل مشتری شکل گرفته است.

معمولاً این مدل در پروژه های بسیار کوچک مورد استفاده قرار میگیرد و تیم تولید نیز از تعدادی اندک تشکیل گردیده است.

### طرح و کاربرد مدل بیگ بنگ

در این مدل کلیه منابع موجود جهت تولید و برنامه نویسی مورد استفاده قرار میگیرد و تمامی توجهات معطوف تولید شده و توجه اندکی به برنامه ریزی میشود. نیازمندیها زمانی که مطرح میگردند، مورد توجه قرار گرفته و پیاده سازی میشوند. اعمال هر گونه تغییر، موجب بازنویسی برنامه نمیگردد.

این مدل برای پروژه های کوچک و همکاری یک یا دو برنامه نویس بسیار ایده آل بوده و برای انجام پروژه های دانشگاهی نیز مطلوب میباشد. کاربرد این مدل برای محصولاتی که نیازمندیهای آن بخوبی مشخص نشده و تاریخی نیز برای نسخه نهائی تعیین نگردیده نیز مدلی مناسب و ایده آل است.

### موافقین و مخالفین مدل بیگ بنگ

یکی از مزایای این مدل سادگی آن است و نیاز اندکی به برنامه ریزی دارد. مدیریت این مدل بسیار سهل و نیازمند هیچ رویه خاصی نیست. بهر حال مدل بیگ بنگ مدلی بسیار پر ریسک بوده و تغییر در نیازمندیها و یا عدم درک نیازها میتواند حتی منجر به نقض یا از بین رفتن کل پروژه شود. این مدل صرفاً جهت پروژه های کوچک و تکراری که دارای حداقل ریسک میباشد مناسب است.



جدول زیر نشاندهنده نظرات موافق و مخالف این مدل میباشد.

نظرات موافق	نظرات مخالف
<ul style="list-style-type: none"> <li>• مدلی بسیار ساده است .</li> <li>• نیاز اندک به برنامه ریزی دارد.</li> <li>• مدیریت آن آسان است.</li> <li>• منابع اندکی مورد نیاز میباشد.</li> <li>• قابلیت انعطاف بسیاری برای برنامه نویسان دارد.</li> <li>• کمک آموزشی مناسبی برای دانشجویان و تازه واردین است.</li> </ul>	<ul style="list-style-type: none"> <li>• مدلی نامطمئن و پر ریسک است .</li> <li>• برای پروژه های بزرگ و موجودیت گرا مدل مناسبی نیست.</li> <li>• مدلی ضعیف برای پروژه های طولانی و مداوم است.</li> <li>• در صورت عدم درک نیازمندیها، میتواند بسیار پر هزینه باشد.</li> </ul>

## 6: چرخه حیات تولید نرم افزار - مدل چابک Agile Model

مدل چابک در چرخه حیات تولید نرم افزار ترکیبی از مدل تکراری و مدل آبشاری است که بر توافق و رضایت مشتری توجه داشته و بر تحویل سریع محصول نرم افزاری به مشتری تاکید دارد.

در مدل چابک روند تولید محصول به مراحل کوچک تقسیم میگردد و مراحل بصورت مکرر ادامه می یابد. مراحل تکراری هر یک در هر دوره یک الی سه هفته به درازا می کشد و در مرحله تکراری تیم های کاری بصورت همزمان بر در حوزه های مختلف نظیر برنامه ریزی، تجزیه و تحلیل نیازمندیها، طراحی، کد نویسی، آزمون دستگاه و آزمون قبولی کار میکنند.

در پایان مراحل تکراری، محصول به مشتریان و ذینفع های عمده نمایش و ارائه میگردد.

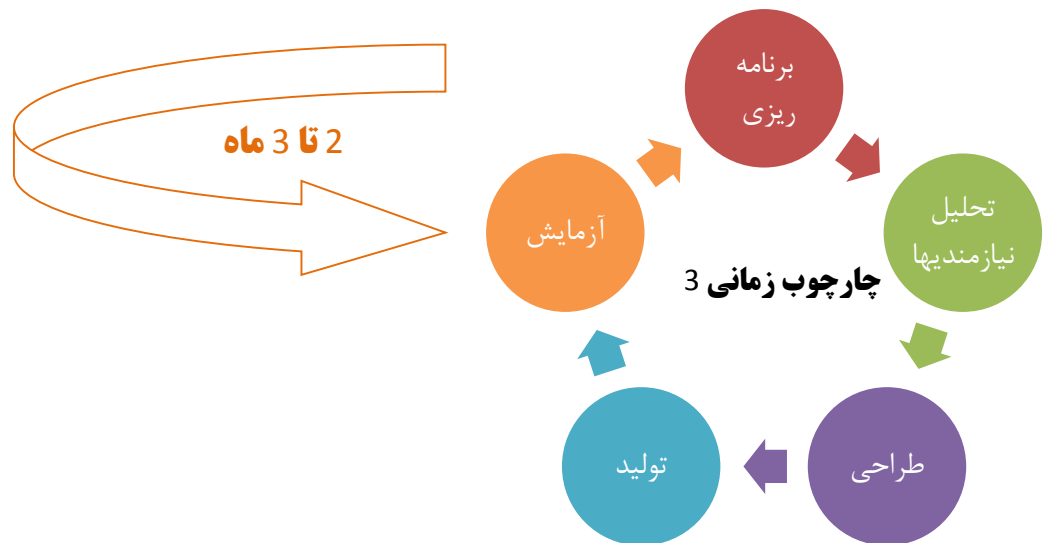
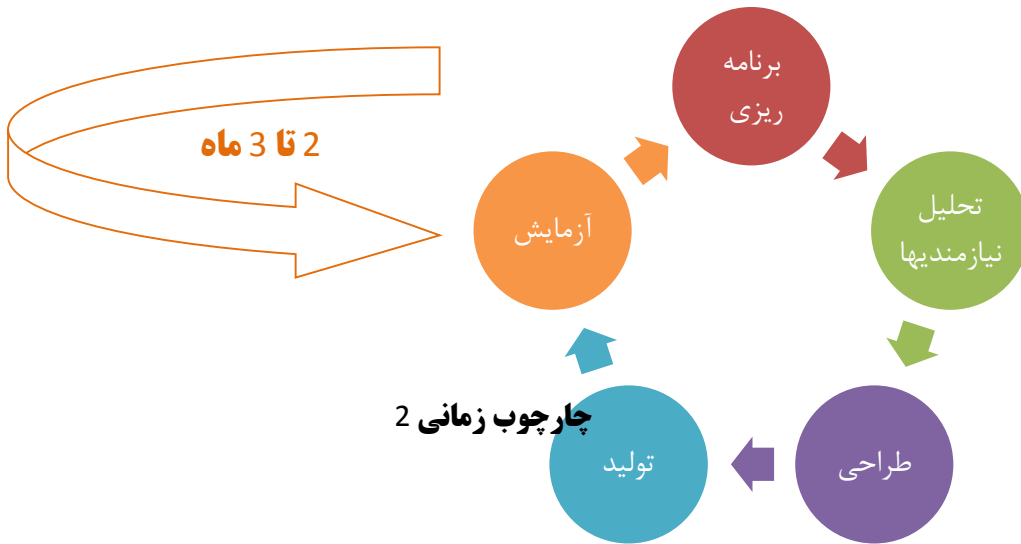
مدل چابک چیست؟

گروههایی که پروژه تولید نرم افزار را از طریق مدل چابک انجام میدهند، اعتقاد دارند که هر پروژه ای را باید بصورت مختلف اجرا نمود و روشهای موجود را باید با مناسب ترین شکل با نیازمندیهای پروژه انطباق داد.

در مدل چابک، وظایف به ظرفیت های زمانی (چهارچوب های زمانی کوتاه) تقسیم میگردد تا در این زمانها بتوان ویژگیها و خصیصه های را جهت انتشار و تهیه نسخه ای از محصول بدست آورد.

اتخاذ شیوه تکراری در مدل چابک باعث میگردد که پس از هر مرحله تکرار، بخشی از نرم افزار تحویل گردد. هر یک از بخشها از نظر ویژگیها حالت تصاعدی و افزایشی داشته و بخش نهائی دارای کلیه ویژگی های مورد نیاز مشتری میباشد.

تصویر ذیل نشاندهنده یک مدل چابک میباشد.



روش مدلسازی چابک در تولید محصولات نرم افزاری بسیار زود آغاز گردید و با گذشت زمان بلحاظ داشتن قابلیت انعطاف و سازگاری، شهرت بسیار پیدا کرد. مشهورترین مدلها و روشهای چابک عبارتند از:

- Rational unified process (1994)
- Scrum (1995)
- Crystal clear Extreme programming (1996)
- Adaptive software Development (ASD)
- Feature Driven Development (FDD)
- Dynamic system Development Method (DSDM) (1995)

پس از انتشار بیانیه ای در سال 2001، کلیه روشهای فوق را اکنون بعنوان متدولوژی چابک میخوانند.

**این بیانیه شامل چهار اصل میباشد :**

- افراد و فعل و انفعالات – در روش چابک، خود سازمان دهی و انگیزه بهمان میزان تعامل و همکاری درون گروهی و موقعیت و همچنین برنامه نویسی گروهی دارای اهمیت است.
- نرم افزار کارآمد – نرم افزار نمایشی کارآمد بعنوان بهترین وسیله ارتباطی با مشتری در جهت شناخت هر چه بیشتر نیازهای وی محسوب میگردد و این شیوه به ارائه مستندات ارجحیت دارد.
- همکاری مشتری – از آنجائی که شناخت نیازمندیها بنابر عوامل متعدد در ابتدای شروع پروژه بصورت کامل قابل جمع آوری نمیباشد، لذا همکاری متقابل و مداوم مشتری در جهت بدست آوردن نیازمندیهای محصول از اهمیت خاصی برخوردار میباشد.
- واکنش در قبال تغییرات – مدل چابک دارای قابلیت پاسخگویی سریع به تغییرات و تولید مداوم دارد.

### **مدل چابک در قبال مدلهای چرخه حیات تولید نرم افزار**

مدل چابک مدلی مبتنی بر روش تولید تطبیقی نرم افزار است در حالیکه مدلهای سنتی نظیر مدل آبشاری مبتنی بر دیدگاه فرضیات از پیش تعیین شده میباشد.

گروه یا تیمی که با روشهای سنتی سروکار دارند معمولاً از روی برنامه ای که جزئیات آن بدقت تعیین گردیده کار کرده و کلیه وظایف به دقت کامل پیش بینی گردیده و زمان تحویل محصول نیز در چند ماه بعد و یا در طول چرخه حیات مشخص و معین گردیده است.

این مدلها کاملاً بستگی به تجزیه و تحلیل نیازمندیها و برنامه ریزی هائی دارد که در آغاز چرخه حیات انجام گرفته است.

چنانچه در پروژه ای که با مدل‌های سنتی انجام میگیرد تغییری گنجانده شود مدیریت کنترل تغییرات، الویت اعمال آنرا پس از درخواست تغییر مورد بررسی قرار میدهد.

در مقابل در مدل چابک که از دیدگاه انطباقی تبعیت مینماید هیچ گونه برنامه ریزی مشروح و دقیق وجود نداشته و وظایف آتی زمانی مشخص و روشن میگردند که ویژگیهای پروژه نیاز به تغییر و یا تولید داشته باشند. در این مدل، تولید براساس ویژگیها انجام و تیم یا گروه کاری خود را بصورت پویا با نیازمندی مربوط به ایجاد تغییر در محصول وفق میدهند. محصول، مکرراً در هر مرحله آزمایش و این مسئله ریسک های ناشی از عدم موفقیت های آتی را به حداقل میرساند.

همکاری متقابل مشتری با گروه کاری، ستون فقرات متدولوژی چابک محسوب میگردد و ارتباط نزدیک همراه با تهیه حداقل مستندات، ویژگی بارز محیط تولید چابک را تشکیل میدهد.

افراد گروه کاری در مدل چابک تشریک مساعی کامل با یکدیگر داشته و در اغلب اوقات در یک موقعیت جغرافیائی با یکدیگر همکاری میکنند.

### موافقین و مخالفین مدل چابک

مدل چابک در سالهای اخیر بصورت عمده در جهان نرم افزار پذیرفته شده است ولی این روش نمیتواند برای تولید کلیه نرم افزارها روشی مطلوب محسوب گردد.

در جدول ذیل نظرات موافقین و مخالفین مدل چابک مشاهده میگردد.

نظرات موافق	نظرات مخالف
<ul style="list-style-type: none"> <li>دیدگاهی بسیار واقع گرایانه در تولید نرم افزار است.</li> <li>کارگروهی را ترویج داده و آموزش متقابل را ارتقا می بخشد.</li> <li>کارکردها به سرعت تکامل یافته و اثبات میگردد.</li> <li>نیاز به منابع بسیار تقلیل می یابد.</li> <li>برای نیازمندیهای ثابت و یا متغییر روشی مطلوب میباشد.</li> <li>راه حل‌های عملی مقدماتی سریعاً تحویل میشوند.</li> <li>برای محیط‌هایی که دائماً در حال تغییر میباشند مدلی مناسب بشمار میرود.</li> <li>با پیروی از کمترین قوانین و مقررات، مستندات به سادگی تهیه میگرددند.</li> <li>با تولید یا تولید همزمان، تحویل محصول در زمان برنامه ریزی شده میسر میگردد.</li> <li>نیاز به برنامه ریزی نداشته و یا به حداقل آن نیاز دارد.</li> <li>مدیریت آن آسان است.</li> <li>برای برنامه نویسان قابلیت انعطاف ایجاد مینماید.</li> </ul>	<ul style="list-style-type: none"> <li>جهت اجرای پروژه های پیچیده مطلوب و مناسب نیست .</li> <li>تداوم، حفظ مراحل و تولید پذیری آن با ریسک توأم میباشد.</li> <li>داشتن برنامه کلی، مدیری چابک جهت انجام عملیات مدیریت پروژه ضروری میباشد که بدون آن این مدل بی فایده خواهد بود.</li> <li>مدیر پروژه، حوزه عملیات، عملکرد و تعدیل و تغییر ضرب العجل ها را تعیین مینماید.</li> <li>بستگی کامل به فعل انفعال مشتری دارد و چنانچه مشتری فاقد آگاهی باشد تیم پروژه میتواند مسیری نادرست را طی کند .</li> <li>این مدل بستگی بسیار به فرد داشته و بنابراین حداقل مستندات تهیه میگردد.</li> </ul>

- انتقال فن آوری به اعضای جدید تیم احتمالاً بنا به دلیل کمبود مستندات میتواند بسیار چالش برانگیز باشد.

### 7: چرخه حیات تولید نرم افزار - مدل تولید سریع برنامه کاربردی

#### Rapid Application Development (RAD)

مدل تولید سریع برنامه کاربردی، مدلی است مبتنی بر نمونه سازی و تولید تکراری بدون برنامه ریزی مشخص و پیچیده. روند نگارش نرم افزار، خود شامل برنامه ریزی مورد نیاز جهت تولید سیستم میباشد. مدل تولید سریع بیشتر متمرکز بر جمع آوری نیازمندیهای مشتری از طریق ایجاد کارگاه و یا گروهها دارد. در این مدل سپس نمونه سازی هائی را که مشتریان به شیوه تکراری و یا استفاده مجدد از نمونه های موجود تولید نموده اند را آزمایش و پس از ادغام نسبت به تحویل سریع آن اقدام میگردد.

#### RAD چیست؟

مدل تولید سریع یک روش یا شیوه تولید نرم افزار است که از کمترین برنامه ریزی بمنظور نمونه سازی سریع استفاده مینماید. نمونه سازی یک مدل کاری است که از نظر عملکرد مشابه اجزای ترکیب دهنده یک محصول میباشد. در مدل تولید سریع، پیمانانه های تابعی یا عملکردی بصورت موازی و بعنوان یک نمونه تهیه و جهت تحویل سریع محصول، پیمانانه ها با یکدیگر ادغام تا یک محصول کامل ساخته شود.

از آنجائی که در این مدل هیچ برنامه ریزی مشروحوئی از پیش انجام نگرفته است لذا اعمال تغییرات در حین روند تولید آسانتر میباشد.

پروژه هائی که با مدل تولید سریع انجام میگیرند روشهای تکراری و افزایشی را پیروی کرده و گروههای کوچکی مرکب از برنامه نویسان، متخصصان، نمایندگان مشتری و سایر منابع فن آوری اطلاعات، بتدریج در بخش یا مولفه و یا نمونه سازی خود فعالیت میکنند.

مهمترین جنبه ای که این مدل را میتواند موفق نماید این است که از قابلیتهای استفاده میتوان مجددا استفاده نمود.

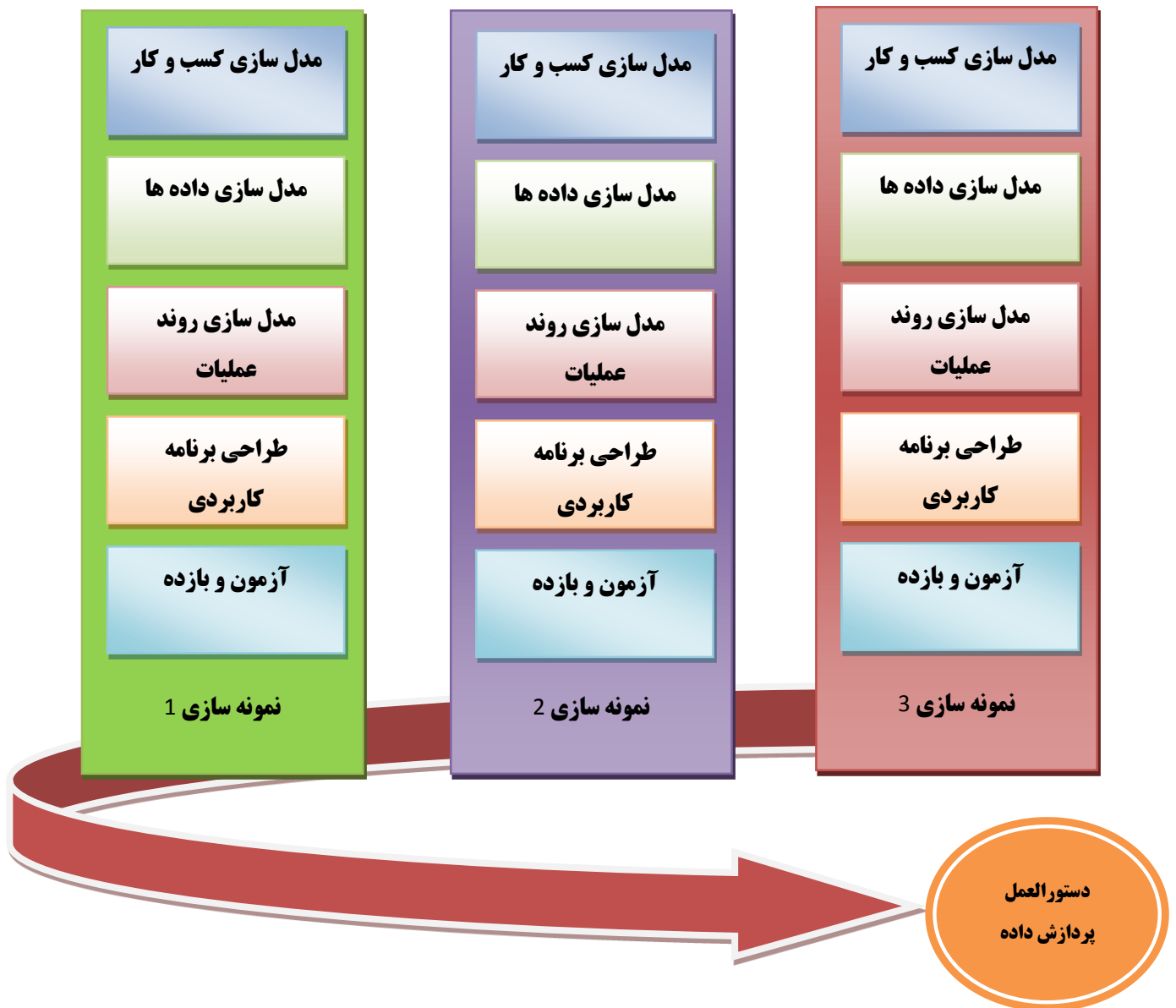
## طرح مدل تولید سریع

مدل تولید سریع مراحل تجزیه و تحلیل، طراحی، تهیه و آزمون را به یک سری از چرخه های تولید تکراری و کوتاه مدت تقسیم مینماید. در زیر مراحل مدل تولید سریع تشریح گردیده است:

- مدل سازی کسب و کار: مدل کسب و کار برای محصولی که در حال تهیه و تولید میباشد بصورتی طراحی میشود که گردش اطلاعات و توزیع اطلاعات را بین ورودی و خروجیهای گوناگون داد و ستد مشخص نماید. برای یافتن اطلاعات حیاتی کسب و کار یا دادو ستد و نحوه بدست آوردن آن و چگونگی و زمان پردازش اطلاعات و آگاهی از عوامل موفقیت گردش اطلاعات می باید یک تجزیه و تحلیل کامل انجام گیرد.
- مدل سازی داده ها: اطلاعاتی که در مرحله مدل سازی کسب و کار گردآوری شده اند برای شکل دادن به مجموعه ای از اشیای داده ای که برای انجام کسب و کار حیاتی هستند مورد بررسی و تجزیه و تحلیل قرار میگیرند. مشخصات هر یک از مجموعه داده ها، شناسائی و مشخص میگردند. رابطه بین اشیای داده ای تعیین و ارتباط آن با مدل کسب و کار مشخص میگردد.
- مدل سازی روند عملیات: مجموعه اشیای داده ای که در مرحله مدل سازی داده ها تعریف شده اند جهت فراهم نمودن جریان و گردش اطلاعات مود نیاز برای نیل به اهداف مشخص کسب و کار تغییر و تبدیل می یابند.
- روند مدل سازی برای هرگونه تغییر و یا بهبود در مجموعه اشیای داده ای در این مرحله تعریف میگردند. در این مرحله هرگونه اضافه، حذف، اصلاح و بازیابی و یا تغییر در اشیای داده ای انجام می پذیرد.
- تولید برنامه کاربردی: در این مرحله سیستم واقعی ایجاد و عمل رمز گذاری نیز با استفاده از ابزار اتوماسیون جهت تبدیل مدل سازی داده ها به یک نمونه واقعی انجام میگردد.
- آزمون و بازده: آزمون همه جانبه و تبدیل مدل تولید سریع به یک نمونه اصلی در طی هر یک از دوره های تکراری انجام میگردد. جریان گردش داده ها و همچنین ارتباط بین کلیه اجزا بصورت کامل تحت آزمایش در میآیند. با این روند چون اغلب مولفه های برنامه نویسی آزمایش شده اند بنابراین میتوان گفت که این امر باعث کاهش و تقلیل ریسکهای عمده خواهد شد.



تصویر مدل تولید سریع در زیر مشاهده می‌گردد.



## مدل تولید سریع در مقابل مدل‌های سنتی چرخه حیات تولید نرم افزار

مدل‌های سنتی معمولاً از روند انعطاف ناپذیر پیروی مینمایند و بیشتر آنها بر تجزیه و تحلیل نیازمندیها و جمع آوری آنها قبل از شروع برنامه نویسی تاکید می‌ورزند. در این مدلها بر مشتری فشار آورده میشود که نیازمندیهای خود را قبل از شروع پروژه تصریح و تأیید و امضا نماید و مشتری نظر به اینکه برای مدتی طولانی چون محصول خاصی تولید نشده احساسی نسبت به آن نخواهد داشت.

مشتری پس از اینکه عملاً نرم افزار را مشاهده نمود امکان دارد خواهان اصلاح یا تغییری در نرم افزار گردد ولی روند ایجاد تغییر در این مدل‌های سنتی کاملاً سخت و توأم با اشکال بوده و امکان اعمال تغییرات عمده در این مدلها عملی نباشد. برخلاف مدل های سنتی، مدل تولید سریع جهت تحویل مدل‌های کارا به مشتری از شیوه تکراری و افزایشی پیروی مینماید.

نتیجه این امر تحویل سریع نرم افزار به مشتری بوده و مشتری در جریان تولید و در کل چرخه حیات آن نیز دخیل بوده است لذا در این مدل ریسک عدم انطباق مشخصات نرم افزار با نیازمندیهای عملی کاربر کاملاً کاهش میابد.

## کاربرد مدل تولید سریع

مدل تولید سریع را میتوان بصورت موفقیت آمیز در پروژه هایی که دارای واحد ها و پیمانانه های آشکار میباشد به کار بست اگر پروژه را نتوان به پیمانانه های مختلف تقسیم نمود، مدل تولید سریع میتواند با عدم موفقیت روبرو شود.

مدل تولید سریع را میتوان در سناریوهای ذیل بکار بست:

- مدل تولید سریع را صرفاً میتوان در سیستم هایی بکار برد که قابل تقسیم به واحد ها و پیمانانه های کوچکتر باشد و امکان تحویل آن بصورت افزایشی میسر میگردد.
- این مدل در صورتی قابل استفاده است که دسترسی به طراحان مدل سازی فراهم باشد.
- این مدل صرفاً در مواردی قابل استفاده است که بودجه لازم جهت کاربرد ابزار تولید بصورت کامل تامین شده باشد.
- این مدل صرفاً در مواردی میتوان انتخاب کرد که متخصصان این حوزه با دانش مناسب و کافی حضور داشته باشند.
- این مدل را میتوان در مواردی بکار برد که نیازمندیها در طول دوره پروژه تغییر یابد و نمونه تهیه شده در دوره های کوتاه دو تا سه ماه به مشتری ارائه گردد.

## نظرات موافقین و مخالفین مدل تولید سریع

مدل تولید سریع قادر به ارائه سریع محصول میباشد زیرا در این مدل زمان تولید بلحاظ قابلیت استفاده مجدد اجزا و همچنین تولید و تهیه موازی کاهش میابد.

مدل تولید سریع زمانی قادر به کار به نحو بهینه میباشد که مهندسین ماهر در دسترس باشند و مشتری نیز خواهان نیل به نمونه مورد نظر در چهارچوب زمانی تعیین شده باشد. اگر این دو مورد در هر دو سو وجود نداشته باشد مدل محکوم به شکست خواهد بود.

در ذیل دلایل موافقین و مخالفین مدل تولید سریع ملاحظه میشود:

نظرات موافق	نظرات مخالف
<ul style="list-style-type: none"> <li>• مدل را میتوان با تغییر نیازمندیها سازگار نمود.</li> <li>• میزان تولید را میتوان اندازه گیری کرد.</li> <li>• زمان تکرار عملیات را میتوان با استفاده از ابزار قدرتمند مدل تولید سریع کاهش داد .</li> <li>• بهروری را با افراد کمتر و زمان کوتاهتر افزایش داد .</li> <li>• زمان تولید را میتوان تقلیل داد.</li> <li>• استفاده مجدد از اجزا و مولفه ها را افزایش داد.</li> <li>• بررسی سریع مراحل امکان پذیر میباشد.</li> <li>• میتوان مشتری را نسبت به ارائه نظریات دلگرم نمود.</li> <li>• مسائل ادغام موارد را از همان آغاز عملیات حل نمود.</li> </ul>	<ul style="list-style-type: none"> <li>• وابستگی کامل به اعضای تیم قوی فنی شناخت نیازمندیها را امکان پذیر میسازد.</li> <li>• صرفاً سیستمهای دارای واحد و پیمانها میتواند از این مدل استفاده نمایند.</li> <li>• این مدل نیاز شدید به برنامه نویسان و طراحان ماهر دارد.</li> <li>• این مدل وابستگی کامل به مهارتهای مدل سازی دارد.</li> <li>• این مدل وابستگی کامل به مهارتهای مدل سازی دارد.</li> <li>• این مدل برای پروژه های ارزان مناسب نیست زیرا هزینه مدل سازی و تولید اتوماتیک که بسیار بالا است .</li> <li>• پیچیدگی مدیریت بیش از سایر مدلها است .</li> <li>• این مدل برای سیستمهای مبتنی بر اجرای جداگانه و مقیاس پذیر مطلوب میباشد.</li> <li>• این مدل حضور و درگیری کاربر در چرخه حیات را نیاز دارد.</li> <li>• این مدل برای پروژه هایی که نیاز به زمان کوتاه تر دارد مناسب است.</li> </ul>

## فشرده راهنمای آموزشی

این راهنمای آموزشی حاوی مدل‌های گوناگون چرخه حیات تولید نرم افزار و سناریوهائی است که مدل‌های مختلف را مورد استفاده قرار می‌دهند. اطلاعات این راهنمای آموزشی مدیران پروژه را در انتخاب مدل مناسب پروژه یاری رسانده و همچنین برنامه نویسان و تست کنندگان را کمک مینماید که از اصول اساسی مدل‌های تولید آگاهی یابند.

در این راهنما کلیه مدل‌های چرخه حیات تولید و یا تولید نرم افزار اعم از مدل‌های سنتی و یا مدرن بررسی گردیده و در پایان هر مرحله نظرات موافق و مخالف و راههای کاربرد عملی مدل‌های چرخه حیات تولید نرم افزار نیز مورد ارزیابی قرار گرفته است.

مدل آبشاری و مدل V مدل‌های سنتی و از نوع افزایشی هستند. منظور از افزایشی این است که مرحله بعدی صرفاً میتواند پس از تکمیل مرحله قبلی آغاز گردد. این مدلها برای پروژه هائی مطلوب هستند که نیازمندیهای محصول روشن و آشکار بوده و این نیازها در طول دوره و تا تکمیل پروژه تغییر نخواهد کرد.

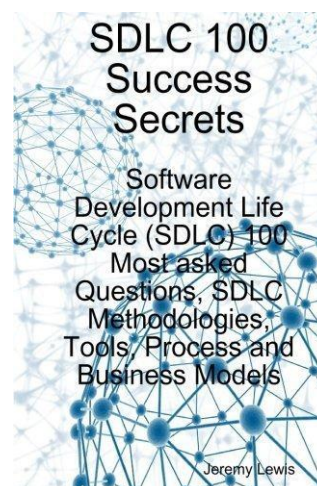
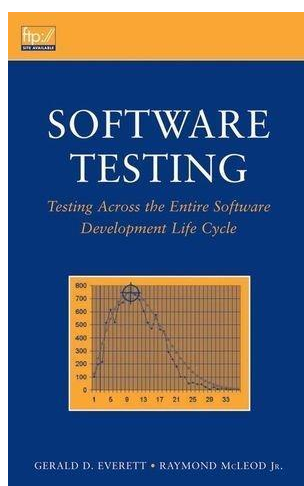
مدل‌های تکراری یا حلزونی با تغییرات سازگار بوده و برای پروژه هائی که نیازمندیها بنحوی تعریف و مشخص نشده اند و یا نیازمندیهای بازار غالباً دچار تغییر میگردند مناسب میباشدند.

مدل بیگ بنگ از زمره روشهای نادر در تولید نرم افزار بوده و صرفاً جهت پروژه های کوچک و یا آکادمیک مطلوب میباشد.

مدل چابک معروف ترین مدل در صنعت محسوب میگردد. مدل چابک با استفاده از شیوه مدل سازی توان تحویل سریع محصول به مشتری را دارا میباشد. مدل چابک پروژه را به بخشهای تکراری کوچک که هر یک دارای ویژگی خاصی است تقسیم مینماید. تعامل مشتری، در این مدل ستون فقرات متدولوژی را تشکیل داده و ارتباط باز با حداقل مستندات از مشخصات بارز محیط تولید چابک محسوب میگردد.

مدل تولید سریع و نمونه سازی نرم افزار، تکنیکی نوین در جهت شناخت نیازمندیها در مراحل اولیه چرخه پروژه بشمار میرود. در این مدل و تکنیک، یک مدل نمونه و کاری جهت مشاهده و ارائه نظریات به مشتریان و ذینفعان ارائه میگردد. بازخورد مشتریان با شیوه ای منظم و متشکل جهت اصلاح و تکمیل محصول مورد استفاده قرار میگیرد.

Books	Websites
Learn Software Development: An Agile Toolkit for Software Development Managers – by Mary Poppendieck, Tom Poppendieck, Ken Schwaber	<a href="http://en.wikipedia.org/wiki/systems_Development_Life_Cycle">http://en.wikipedia.org/wiki/systems_Development_Life_Cycle</a>
The Art of Software Testing-By Glenford J Myers	<a href="http://en.wikipedia.org/wiki/Agile_software_development">http://en.wikipedia.org/wiki/Agile_software_development</a>
Agile Project Management eith Scrum – by Ken Schwaber	<a href="http://agilemanifesto.org">http://agilemanifesto.org</a>
Extreme Programming Explained – Book by Kent Beck	<a href="http://www.agilealliance.org">http://www.agilealliance.org</a>



The Software Development Lifecycle  
A complete guide  
By Richard Murch

