



چرخه حیات توسعه نرم افزار: راهنمایی کامل برای ارتقا و بهبود پروژه های نرم افزاری

فهرست

- 3 چرخه حیات توسعه نرم افزار چیست؟
 - 3 چرخه حیات توسعه نرم افزار چگونه ایجاد شد؟
 - 3 چرا چرخه حیات توسعه نرم افزار مهم است؟
 - 4 نقش امنیت در چرخه حیات توسعه نرم افزار
 - 5 چرخه حیات توسعه نرم افزار چگونه کار می‌کند؟
 - 6 بهترین شیوه‌های چرخه حیات توسعه نرم افزار
- نتیجه گیری 6

ساختن یک نرم‌افزار یک فعالیت بزرگ است و به همین دلیل، تیم‌های محصولات دیجیتال بر روی چرخه حیات توسعه نرم‌افزار (SDLC) تکیه می‌کنند. توسعه نرم‌افزار یک فرآیند تکراری است که برای پروژه‌های نرم‌افزاری که از چندین مرحله برای ساخت و اجرای برنامه‌های نرم‌افزاری تشکیل شده‌اند، پیروی می‌شود. SDLC به بهبود و اندازه‌گیری فرآیند کمک می‌کند و امکان تحلیل و بررسی توسعه نرم‌افزار در هر مرحله را فراهم می‌کند. چرخه حیات توسعه نرم‌افزار (SDLC) اصطلاحی است که در صنعت نرم‌افزار برای توصیف فرآیند ایجاد یک محصول نرم‌افزاری جدید استفاده می‌شود. برنامه نویسان نرم‌افزار از آن به عنوان یک راهنما استفاده می‌کنند تا مطمئن شوند نرم‌افزار با هزینه‌های کمتر و کیفیت بالاتر در زمان کمتری تولید شود.

چرخه حیات توسعه نرم افزار چیست؟

چرخه عمر توسعه نرم‌افزار (SDLC) یک فرآیند ساختار یافته است که امکان تولید نرم‌افزار با کیفیت بالا و کم هزینه را در کمترین زمان ممکن فراهم می‌کند. هدف SDLC تولید یک نرم‌افزار برتر است که تمام انتظارات و خواسته‌های مشتری را برآورده کند و از آن فراتر هم برود. چرخه حیات توسعه نرم افزار یک طرح تفصیلی را با مراحل که هر کدام شامل فرآیندها و تولیدات مشخصی هستند، تعریف و ترسیم می‌کند. پایبندی به SDLC سرعت توسعه را افزایش می‌دهد و خطرات پروژه و هزینه‌های مرتبط با روش‌های جایگزین تولید را به حداقل می‌رساند.

چرخه حیات توسعه نرم افزار چگونه ایجاد شد؟

در دهه‌های ۱۹۵۰ و ۱۹۶۰، علم کامپیوتر به سرعت پیشرفت کرد. این پیشرفت سریع منجر به ایجاد یک چارچوب تولید شد که در نهایت به توسعه چرخه حیات نرم‌افزار (SDLC) که امروزه می‌شناسیم، تبدیل شد. قبل از دهه ۱۹۵۰، محاسبات به اندازه کافی پیچیده نبودند تا نیاز به یک رویکرد دقیق مانند SDLC داشته باشند. با افزایش پیچیدگی و مقیاس برنامه نویسی، مفهوم برنامه‌نویسی ساختاریافته ظهور کرد. با گذشت زمان، برنامه‌نویسی ساختاریافته نیاز به مدل‌های توسعه تاکتیکی بیشتری داشت، بنابراین SDLC آغاز شد.

چرا چرخه حیات توسعه نرم افزار مهم است؟

SDLC یک چارچوب استاندارد ارائه می‌دهد که فعالیت‌ها را تعریف می‌کند.

- به برنامه ریزی، برآورد و زمان بندی پروژه کمک می‌کند.
- ردیابی و کنترل پروژه را آسان‌تر می‌کند.
- باعث افزایش دید در تمام جنبه‌های پروژه برای همه ذینفعان می‌شود.
- به سازمان‌ها کمک می‌کند تا پیشرفت خود را نسبت به اهداف خود اندازه گیری کنند و مطمئن شوند که همه چیز در مسیر درست قرار دارد.
- سرعت توسعه را افزایش می‌دهد.
- روابط مشتری را بهبود می‌بخشد.
- اشتباهات پروژه را کاهش می‌دهد.
- هزینه‌های مدیریت پروژه و هزینه کلی تولید را کاهش می‌دهد.

نقش امنیت در چرخه حیات توسعه نرم افزار

مفهوم اولیه و ایجاد SDLC در واقع به فعالیت‌های امنیتی به عنوان یک فرایند جداگانه و منحصر به فرد توجه نکرده است که به عنوان بخشی از مرحله آزمایش انجام می‌شود. کاستی‌های این رویکرد پس از وقوع واقعیت، تعداد زیادی آسیب‌پذیری یا باگ که بسیار دیر در فرآیند کشف می‌شدند یا در برخی موارد اصلاً کشف نمی‌شدند، به وجود آمدند. امروزه درک شده که امنیت برای یک SDLC موفق حائز اهمیت است و ادغام فعالیت‌های امنیتی در سراسر SDLC به ایجاد نرم‌افزار قابل اعتماد کمک می‌کند. با ترکیب تدابیر امنیتی در مراحل اولیه چرخه عمر توسعه نرم‌افزار، آسیب‌پذیری‌ها به سرعت کشف و کاهش می‌یابند و هزینه‌های بعدی در چرخه عمر کاهش می‌یابد. یک SDLC امن با انجام ارزیابی‌ها و تدابیر امنیتی در تمام مراحل توسعه نرم‌افزار حاصل می‌شود.

با استفاده از ابزارهای مدرن تست امنیت برنامه، ادغام امنیت در سراسر SDLC آسان است. طبق مفهوم "SDLC امن"، اهمیت دارد که فعالیت‌های تضمین امنیت مانند: آزمایش نفوذ، مدل‌سازی تهدید، بررسی کد و تحلیل معماری به عنوان بخشی جداگانه از تلاش‌های توسعه انجام شوند.

مزایای اصلی دنبال کردن یک رویکرد SDLC ایمن عبارتند از

- نرم افزار امن‌تر به عنوان امنیت یک نگرانی مداوم است
- آگاهی از ملاحظات امنیتی توسط ذینفعان
- تشخیص زود هنگام نقص در سیستم
- کاهش هزینه در نتیجه تشخیص زود هنگام و حل مشکلات
- کاهش کلی ریسک‌های تجاری ذاتی برای سازمان

• نرم افزار امن‌تر باید همواره به عنوان یک نگرانی مداوم در نظر گرفته شود. امنیت نرم افزار نباید فقط در یک مرحله خاص از SDLC مدنظر باشد، بلکه در طول کل فرآیند توسعه و بهره‌برداری، توجه و تلاش مستمری برای بهبود امنیت باید وجود داشته باشد.

• ذینفعان (اشخاص و گروه‌های مختلف که در نرم افزار و سیستم مربوطه دخیل هستند) باید آگاهی کافی در مورد ملاحظات امنیتی داشته باشند. این شامل برنامه نویسان، مدیران، کاربران نهایی و هر کسی که در فرآیند توسعه و استفاده از نرم افزار نقش دارد، می‌شود. آگاهی و آموزش در زمینه امنیت برای ذینفعان کلیدی است.

• تشخیص زود هنگام نقص‌ها و آسیب‌پذیری‌ها در سیستم امری حیاتی است. با استفاده از ابزارها و روش‌های تست امنیت، باید قادر باشیم به سرعت نقاط ضعف و مشکلات امنیتی را تشخیص داده و به آن‌ها رسیدگی کنیم. این کار می‌تواند باعث کاهش خطرات و پتانسیل آسیب‌های بیشتر در سیستم شود.

• تشخیص زود هنگام و حل مشکلات امنیتی منجر به کاهش هزینه‌ها می‌شود. زمانی که مشکلات امنیتی در مراحل اولیه شناسایی شده و بلافاصله برطرف شوند، هزینه‌هایی که ممکن است برای رفع مشکلات بعدی و احتمالی صرف شود، کاهش می‌یابد.

• توجه به امنیت نرم افزار به طور کلی ریسک‌های تجاری ذاتی سازمان را کاهش می‌دهد. حفظ امنیت در سیستم‌ها و نرم افزارها می‌تواند از سوءاستفاده، نشت اطلاعات، خسارت به شبکه‌ها و سیستم‌های مهم سازمان و سایر ریسک‌های امنیتی جلوگیری کند.

چرخه حیات توسعه نرم افزار چگونه کار می‌کند؟

چرخه حیات توسعه نرم افزار (Software Development Life Cycle یا SDLC) یک فرایند مرتبط با توسعه و مدیریت نرم افزار است که در مهندسی نرم افزار استفاده می‌شود. هدف اصلی این چرخه حیات، تشریح و مدیریت فرایندهای مرتبط با توسعه نرم افزار است و شامل مراحل مختلفی می‌شود.

1. برنامه‌ریزی (Planning): در این مرحله، نیازها و اهداف پروژه تعیین می‌شوند و برنامه‌ریزی برای توسعه نرم افزار صورت می‌گیرد. برنامه‌ریزی ممکن است شامل تعیین منابع، زمان‌بندی، و بودجه باشد.
2. تجزیه و تحلیل (Analysis): در این مرحله، نیازها و مشکلات سیستم مورد بررسی قرار می‌گیرند و تحلیل می‌شوند. در این مرحله، نیازمندی‌ها و امکانات سیستم تعیین می‌شوند.
3. طراحی (Design): در این مرحله، طراحی سیستم و نرم افزار صورت می‌گیرد. طرح‌های مختلف مانند طراحی ساختاری و طراحی واسط کاربری به منظور ارائه راه حل‌های مناسب برای نیازمندی‌ها و اهداف پروژه انجام می‌شود.
4. توسعه (Development): در این مرحله، کد نرم افزار تولید می‌شود. برنامه نویسان با استفاده از طرح‌های طراحی شده، کد نرم افزار را پیاده‌سازی می‌کنند.
5. تست (Testing): در این مرحله، نرم افزار توسعه یافته با استفاده از تکنیک‌ها و روش‌های مختلف تست می‌شود تا مشکلات و خطاهای ممکن شناسایی شوند و برطرف شوند.
6. استقرار (Deployment): در این مرحله، نرم افزار آماده برای راه‌اندازی و استقرار به محیط عملیاتی می‌شود. این شامل نصب و پیکربندی نرم افزار در سیستم‌های مقصد است.
7. نگهداری (Maintenance): در این مرحله، نرم افزار به عنوان یک سیستم عمرانی در نظر گرفته می‌شود و تغییرات، بهبودها و تعمیرات لازم را طی می‌کند. همچنین، نیازهای جدید و مشکلات احتمالی مشتریان نیز در نظر گرفته می‌شود.

این چرخه حیات می‌تواند متناسب با نیازها و شرایط پروژه مختلف تغییر کند و به عنوان یک فرایند تکرار شونده در طول عمر نرم افزار مورد استفاده قرار می‌گیرد. همچنین، می‌تواند شامل فرایندهای مرتبط با تامین سخت‌افزار یا ترکیبی از نرم افزار و سخت‌افزار باشد.

بهترین شیوه‌های چرخه حیات توسعه نرم افزار

بهترین روش برای پیاده سازی در SDLC ارتباط موثر بین کل تیم است. هر چه هماهنگی بیشتر باشد، شانس موفقیت بیشتر است.

نشانه‌های یک SDLC به خوبی اجرا شده عبارتند از:

- استقرار موفقیت آمیز یک برنامه جامع امنیتی برنامه
- استانداردهای کیفیت کد
- همکاری موثر بین تیم‌ها
- گردش کار ساده
- مشارکت متقابل تیم‌ها در چرخه حیات توسعه نرم افزار

نتیجه گیری

چرخه حیات توسعه نرم افزار در مهندسی نرم افزار مورد استفاده قرار می‌گیرد و هدف آن تشریح فرایندهای مرتبط با برنامه ریزی، تجزیه و تحلیل، تولید، تست، استقرار و نگهداری سیستم‌های نرم افزاری است که اغلب در یک چرخه، تداوم و تکرار پیدا می‌کند. این چرخه می‌تواند شامل در برگیرنده‌ی فرایندهای مربوط به تامین سخت افزار یا ترکیبی از نرم افزارها و سخت افزارها باشد.

مجله
رهاکو



رهاکو، مرجع تخصصی مجازی سازی ایران

مجله رهاکو

RAHA MAG

آدرس: تهران، خیابان سپهد قرنی، خیابان دهقانی، پلاک 12
کدپستی 1583616414 تلفن: 02154521 www.rahaco.net

