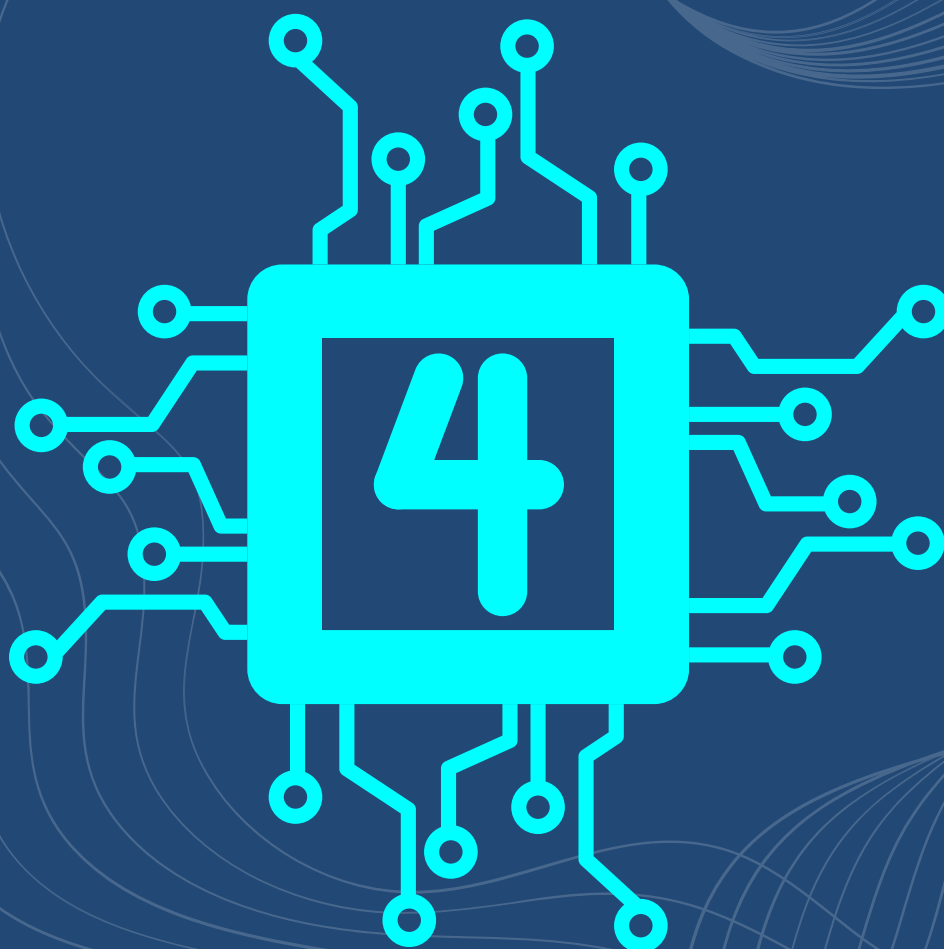


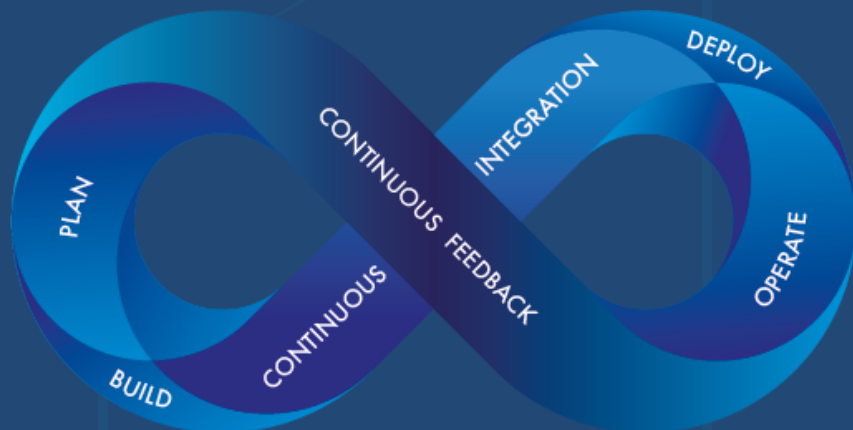
CODE TALE



دوآپس (DEVOPS)

درس برنامه سازی پیشرفته
دکتر مجتبی وحیدی اصل
نشریه دانشکده مهندسی و علوم کامپیوتر

سلام! به قسمت چهارم از CODE TALE خوش آمدید. در این قسمت با مفهوم دواپس (DevOps) و اهمیتش در دنیای مهندسی نرم افزار می‌شویم. به علاوه گریزی به بازار کار می‌زنیم و مسیر تبدیل شدن به یک مهندس دواپس را بررسی می‌کنیم.



مقدمه‌ای بر روش‌های توسعه نرم افزار

فرایند تولید نرم افزار از زمان درخواست مشتری شروع و با گذر از مراحل مانند تعریف پروژه، تعیین و تحلیل نیازمندی‌ها، توسعه و پیاده سازی، تست (Test)، ساخت (Build)، انتشار (Release)، استقرار (Deployment) و در ادامه، دریافت بازخورد و اصلاحات به انتها می‌رسد. البته این فرایند به صورت یک چرخه مداوماً تکرار می‌شود. روش‌ها و متدولوژی‌ها سعی در کوتاه سازی و بهینه کردن این چرخه داشته‌اند. با این وجود، رویکردهای سنتی در مدیریت پروژه، مثل روش Waterfall یا آبشار (یک رویکرد پیشگامانه در چرخه عمر توسعه نرم افزار که با استفاده از آن توسعه نرم افزار به فرایندی خطی تبدیل می‌شود و در آن مراحل و وظایف به طور متوالی قرار می‌گیرند)، به سرعت در حال کهنه شدن و جایگزینی با روش‌های جدیدتر هستند. تیم‌های IT و به طور کلی، شرکت‌ها باید با سرعت و کارایی بیشتری عمل کنند تا بتوانند نرم افزارهای خود را تحویل دهند، رضایت مشتریان خود را جلب نمایند و در رقابت با دیگران پیشرو باشند. در نهایت متدولوژی Agile با هدف چابک سازی این فرایند بوجود آمد ولی تمرکز آن بر نیمی از این فرایند یعنی توسعه (Development) بود. پس از مدتی نقص این چارچوب مبنی بر عدم برقراری ارتباط بین توسعه و عملیات (IT Operations) مشخص شد و از آنجا دواپس (DevOps) از گسترش Agile تا انتهای فرایند بوجود آمد.

دواپس (DevOps) چیست؟

دواپس مجموعه‌ای از روش‌ها، ابزارها و یک فلسفه فرهنگی است که فرایندهای بین تیم‌های توسعه نرم افزار و عملیات IT را به صورت خودکار و یکپارچه می‌کند. این رویکرد بر تقویت تیم، ارتباطات و همکاری‌های متقابل تیمی و اتوماسیون فناوری تأکید دارد.

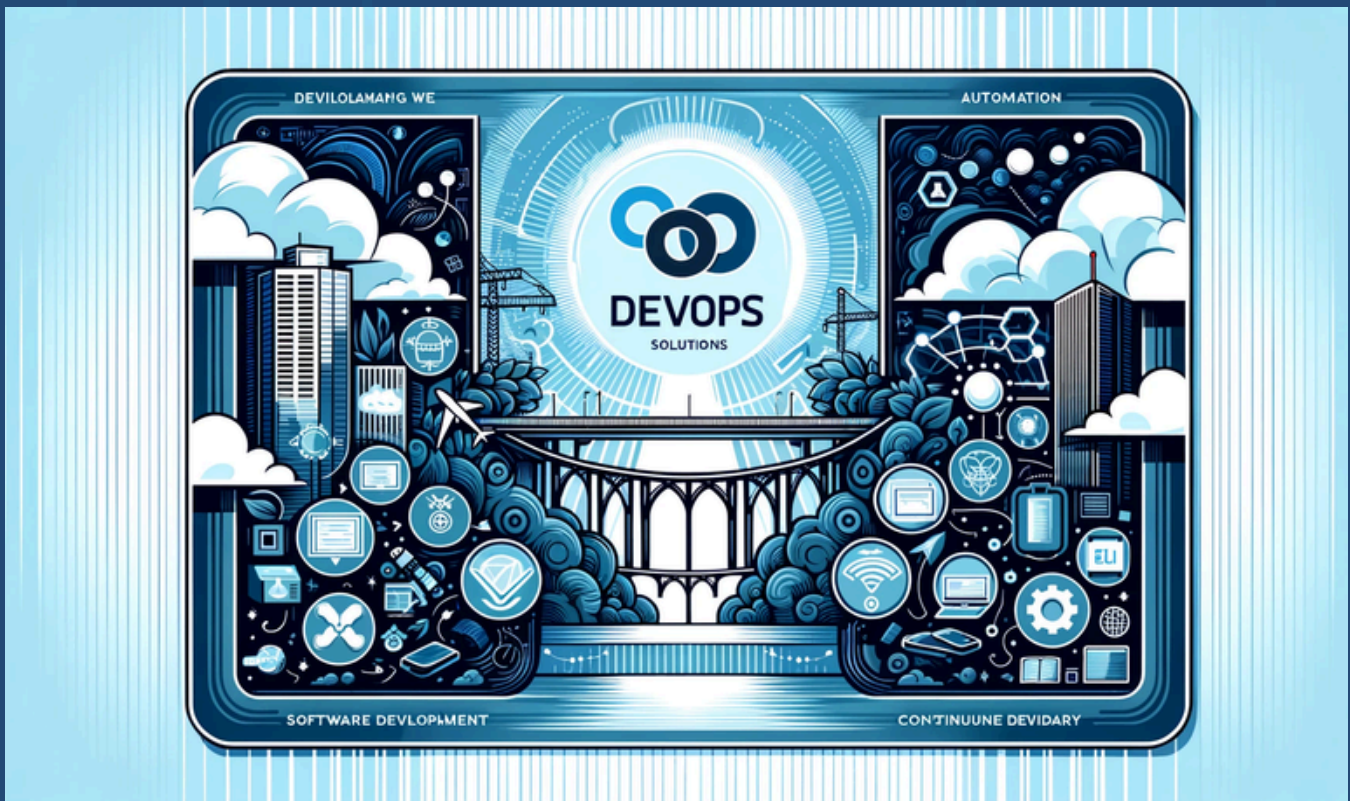
DevOps ترکیبی از توسعه (Dev) و عملیات (Ops) است که به منظور افزایش کارایی، سرعت و امنیت توسعه نرم افزار در مقایسه با فرایندهای سنتی به کار گرفته می‌شود. نتیجه یک چرخه حیات توسعه نرم افزار چابک‌تر، مزایایی برای کسب و کارها و مشتریان آنها به ارمغان می‌آورد.

جنبش دواپس حدود سال ۲۰۰۷ آغاز شد، زمانی که جوامع توسعه نرم افزار و تیم‌های عملیات IT نگرانی‌هایی در مورد مدل سنتی توسعه نرم افزار مطرح کردند؛ جایی که توسعه دهندگانی که کد می‌نوشتند، جدا از تیم عملیاتی که کد را مستقر و پشتیبانی می‌کردند، کار می‌کردند. اصطلاح دواپس، بازتابی از فرآیند یکپارچه سازی این دو رشته در یک فرایند مداوم است.

دوآپس چه مشکلاتی را حل می‌کند؟

مواعی نظیر ناسازگاری‌های نرم‌افزاری، تأخیر در انتشار، مسائل امنیتی نادیده گرفته شده، ارتباطات ناکارآمد بین تیم‌ها و دشواری در مدیریت و پیکربندی زیرساخت‌های پیچیده IT بخشی از مشکلاتی هستند که سازمان‌ها ممکن است در فرایند تولید نرم‌افزار با آن مواجه شوند. این مشکلات می‌توانند روند توسعه و عرضه محصولات جدید را مختل کرده و تأثیر منفی بر رضایت مشتری بگذارند.

یک پروژه دوآپس، با حداقل‌سازی زمان انتظار، فرایندهای دستی و بررسی‌های طولانی، به سرعت از نیازسنجی به سمت تولید نرم‌افزار فعال حرکت می‌کند. زمان‌های چرخه کوتاه‌تر می‌توانند منجر به تحویل محصول به بازار شوند قبل از این که نیازها تغییر بکنند. زمانی که نیازها به دلیل تغییر انتظارات کاربران و تقاضاهای بازار تغییر کنند، زمان‌های چرخه کوتاه می‌توانند به رساندن بروزرسانی‌ها به بازار سرعت بخشند.



همچنین دوآپس مشکلات ارتباطی و اولویت‌بندی بین تخصص‌های IT را حل می‌کند. برای ساخت نرم‌افزار قابل استفاده، تیم‌های توسعه باید محیط تولید را درک کنند و کد خود را در شرایط واقعی آزمایش کنند. یک ساختار سنتی، بین تیم‌های توسعه و عملیات دیوار می‌کشد. این بدان معناست که توسعه‌دهندگان وقتی کدشان به کارکرد لازم برسد دست از کار می‌کشند و اگر فرایند انتشار و استقرار با مشکل مواجه شود، آنوقت صرفاً بر عهده تیم عملیات است که مشکل را حل کند.

در فرهنگ دوآپس، وقتی مشکلی پیش می‌آید، توسعه‌دهندگان به پاسخ "روی دستگاه من کار می‌کرد" متوسل نمی‌شوند. تغییرات اعمال شده به محصول، کوچک و قابل بازگشت هستند. علاوه بر این، کل تیم تغییرات را درک می‌کند که این امر، مدیریت حادثه را ساده‌تر می‌کند.

با یک فرآیند سریع‌تر از ایده تا نرم‌افزار زنده، شرکت‌ها می‌توانند از فرصت‌های بازار بهره‌برداری کنند. به این ترتیب، دوآپس یک نوع مزیت رقابتی برای کسب‌وکارها محسوب می‌شود.

چرخه حیات دوآپس

چرخه حیات دوآپس حاصل ترکیب دو چرخه Dev و Ops می‌باشد. ابتدا به چرخه Dev یا توسعه نرم‌افزار می‌پردازیم. این چرخه ۳ مرحله دارد:

مرحله برنامه‌ریزی (Plan)

مرحله اول این چرخه، مرحله برنامه‌ریزی است. در این فاز، مدیران، اعضای تیم توسعه، کارکنان عملیاتی IT و همه افرادی که در فرایند تولید نرم‌افزار دخیل هستند، چشم‌اندازها و اهداف پروژه را قبل از آغاز تولید و توسعه نرم‌افزار مشخص می‌کنند.



مرحله ساخت (Build)

در این مرحله توسعه‌دهندگان از سیستم‌های کنترل نسخه (Version Control) نظیر گیت (Git) برای توسعه کدهای منبع استفاده می‌کنند که پشتیبانی عالی از شاخه‌بندی (Branching) و ادغام (Merging) را ارائه می‌دهند. حاصل این مرحله، نرم‌افزار قابل اجرا است.



مرحله تست (Test)

این مرحله در چرخه، یک فرایند کلیدی است که در آن کیفیت نرم‌افزار از طریق اجرای مجموعه‌ای از تست‌ها بررسی و تضمین می‌شود. ادغام مداوم (Continuous Integration یا CI) به چندین توسعه‌دهنده امکان می‌دهد تا به یک مخزن مشترک کد کمک کنند. زمانی که تغییرات کد ادغام می‌شوند، تست‌های خودکار اجرا می‌شوند تا قبل از ادغام، صحت آنها تضمین شود.



چرخه Ops به جنبه عملیاتی فرایند انتشار و استقرار نرم‌افزار اشاره دارد. این چرخه ۴ مرحله دارد:

مرحله استقرار (Deploy)

در این مرحله کدهای نرم‌افزاری جهت اجرا بر روی سرورها قرار می‌گیرند که به آن عمل Deployment یا استقرار می‌گویند. استقرار مداوم (Continuous Deployment یا CD) به تیم‌ها اجازه می‌دهد تا ویژگی‌ها را به صورت خودکار و به طور مکرر به تولید منتقل کنند.



مرحله عملیات (Operate)

این مرحله شامل مدیریت نهایی تحویل خدمات IT به مشتریان است و تمامی فعالیت‌های مربوط به طراحی، پیاده‌سازی، پیکربندی، استقرار و نگهداری تمام زیرساخت‌های IT که خدمات یک سازمان را پشتیبانی می‌کند شامل می‌شود.



مرحله نظارت (Observe)

مرحله نظارت در چرخه دوآپس فرایندی است که در آن تیم‌ها عملکرد نرم‌افزار را در محیط واقعی رصد و تحلیل می‌کنند تا در صورت بروز مشکل در زمانی که محصول فعال است، آنها را شناسایی و رفع کنند.



مرحله بازخورد مستمر (Continuous Feedback)

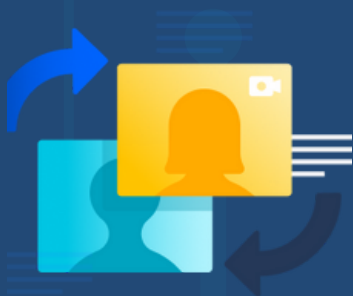
معنای بازخورد مستمر این است که تیم‌های دوآپس باید با ارزیابی هر نسخه انتشار و تهیه گزارش‌هایی، به دنبال بهبود نسخه‌های آتی باشند. از طریق دریافت بازخوردهای پیوسته، آنها قادر خواهند بود تا روندهای کاری خود را تکامل بخشیده و با استفاده از نظرات مشتریان، کیفیت انتشار بعدی را افزایش دهند.



مزایای دواپس

پذیرش دواپس بسیاری از موانع را از بین می‌برد به طوری که تیم‌های توسعه و عملیات به شکلی متصل، راهی کارآمدتر برای فعالیت در تمام چرخه حیات نرم‌افزاری دارند. بدون دواپس، سازمان‌ها اغلب با اصطکاک در انتقال کار بین واحد توسعه و عملیات روبرو می‌شوند که این امر انتشار نسخه‌های نرم‌افزار را به تأخیر می‌اندازد و بر نتایج کسب‌وکار تأثیر منفی می‌گذارد.

مدل دواپس نیاز یک سازمان برای افزایش کارایی عملیاتی، تسریع در تحویل و نوآوری در محصولات است. شرکت‌هایی که فرهنگ دواپس را پیاده‌سازی کرده‌اند از مزایای بهبود همکاری مابین تیم‌ها، قابلیت اطمینان و زمان‌های چرخه کوتاه‌تر بهره‌مند می‌شوند.



بهبود همکاری

پذیرش مدل دواپس، هماهنگی بین تیم‌های توسعه و عملیاتی را ایجاد می‌کند؛ اصطکاک در انتقال کار کاهش یافته و افراد در منافع بیشتری خود را مشترک می‌بینند.

قابلیت اطمینان

استفاده از شیوه‌هایی مانند ادغام مداوم و استقرار مداوم اطمینان حاصل می‌کند که تغییرات کارآمد و ایمن هستند و به بهبود کیفیت محصول نرم‌افزاری کمک می‌کند.



زمان چرخه کوتاه‌تر

ارتباط مکرر بین تیم‌ها زمان چرخه را کوتاه می‌کند؛ کد جدید می‌تواند سریع‌تر منتشر شود در حالی که کیفیت و امنیت حفظ شود. با افزایش سرعت انتشار، تیم‌های دواپس محصولات را به سرعت بهبود می‌بخشند.

چالش‌های دواپس

پارادایم دواپس پیچیدگی‌ها و تغییرات خاص خود را به همراه دارد که ممکن است پیاده‌سازی و مدیریت آنها در یک سازمان پرمشغله دشوار باشد. چالش‌های رایج دواپس شامل موارد زیر است:

- تغییرات سازمانی به خصوص در بخش‌های فناوری اطلاعات، از جمله ایجاد مهارت‌ها و نقش‌های شغلی جدید، که می‌تواند در تیم‌های توسعه و کسب‌وکار اختلال ایجاد کند.
- ایجاد هزینه‌هایی در در راستای نصب، پشتیبانی و آموزش ابزارها و پلتفرم‌ها.
- اتوماسیون‌های گاهی ضعیف پیاده‌سازی شده و ناامن.
- مسائل لجستیکی و سربار (Overhead) ناشی از گسترش دواپس در چندین پروژه و تیم.
- نیاز افراد به رعایت برخی مقررات خاص، بویژه زمانی که جداسازی نقش‌ها ضرورت دارد.

در کل، دواپس هر مشکل کسب‌وکاری را حل نمی‌کند و به یک اندازه به هر پروژه توسعه نرم‌افزار سود نمی‌رساند.

مهندس دوآپس کیست؟

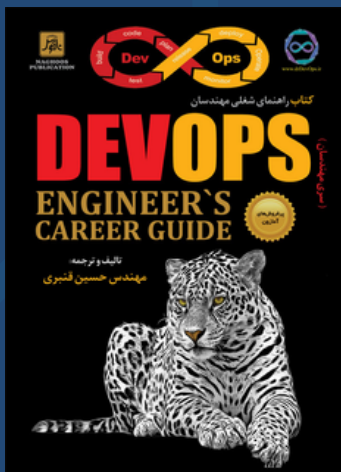
یک مهندس دوآپس مسئول تمام جنبه‌های چرخه حیات توسعه نرم‌افزار، از جمله تسهیل ارتباط میان تیم توسعه و عملیات می‌باشد. آنها با پایبندی به روش‌ها و اصول دوآپس، فرایندهای توسعه را به طور کارآمد و در جریان کار ادغام می‌کنند، اتوماسیون را جایی که امکان‌پذیر است معرفی می‌کنند و کد را آزمایش و تحلیل می‌کنند. همچنین این افراد مسئولیت ایجاد، ارزیابی، مستقرسازی و بروزرسانی ابزارها و پلتفرم‌ها (شامل زیرساخت‌های IT) را به عهده دارند. مهندسان دوآپس انتشارها را مدیریت کرده و تلاش می‌کنند مشکلات فنی را برای کاربران نرم‌افزار شناسایی و به حل آنها کمک کنند.

چگونه مهندس دوآپس شویم؟

مهارت‌های فنی مورد نیاز مهندس دوآپس بسته به ساختار تیم، فناوری‌ها و ابزارهای موجود در سازمان متفاوت است. تبدیل شدن به یک مهندس دوآپس نیازمند علاقه‌مندی به نوشتن کد و حل مشکلات توسعه، مدرک کارشناسی و تجربه عملی است. فهرست زیر مهارت‌ها، وظایف و مسئولیت‌هایی را نشان می‌دهد که انتظار می‌رود یک مهندس دوآپس از عهده آنها برآید:

- مهارت در استفاده و مدیریت سیستم‌های عامل مبتنی بر لینوکس
- مهارت در شناسایی و رفع خطاهای متداول در انواع سیستم‌ها
- طراحی، تجزیه و تحلیل و ارزیابی چالش‌های مرتبط با اتوماسیون سیستم‌ها
- دارا بودن تجربه قابل توجه در کدنویسی با زبان‌های اسکریپتی نظیر پایتون (Python)، روبي (Ruby)، پرل (Perl) و ...
- تسلط بر مفاهیم CI (Continuous Integration) و CD (Continuous Deployment) و استفاده مؤثر از ابزارهایی مانند Maven، Jenkins و ...
- تجربه در کار با سیستم‌های Containerization مانند Docker و Kubernetes (کانتینریزاسیون یک روش مجازی‌سازی در سطح سیستم عامل - و نه سخت افزار - بر روی چند منبع شبکه است که در آن نرم‌افزارها می‌توانند در فضاهای کاربری مجزا به نام Container اجرا شوند).
- آشنایی و مهارت در استفاده از ابزارهای مدیریت پیکربندی مثل Puppet، SaltStack و ...
- تجربه در مدیریت و نظارت بر عملیات با استفاده از ابزارهای مانیتورینگ (Monitoring) و لاگینگ (Logging) مانند Prometheus، Grafana و ELK Stack
- دانش در زمینه شبکه‌های کامپیوتری و امنیت سایبری برای اطمینان از امنیت و قابلیت اطمینان سیستم
- داشتن مهارت‌های ارتباطی و توانایی در یادگیری سریع موضوعات جدید و پیچیده
- مهارت در برقراری ارتباط و همکاری با تیم‌های دیگر برای ایجاد فرآیندهای کاری یکپارچه و مؤثر

معرفی یک کتاب



کتاب "راهنمای شغلی مهندسان DEVOPS" یک مرجع عالی و البته نسبتاً کوتاه برای سوالاتی است که هنوز پاسخ آنها را با خواندن این نشریه راجع به متدولوژی دوآپس نیافتید. در طول مطالب کتاب، شما اطلاعاتی را در مورد اینکه دوآپس چیست و چگونه می‌توانید از آن برای بهبود روند خود استفاده کنید، به دست آورده و نقش‌های متفاوتی را که با دوآپس مرتبط است، خواهید شناخت. چند فصل آخر کتاب شامل اطلاعاتی است در مورد مهارت‌های لازم برای توسعه و مسیری که باید انتخاب کنید. همچنین، فصل آخر شامل نمونه پرسش و پاسخ‌هایی است که به صورت متداول از افراد در طی مصاحبه برای کسب موقعیت شغلی مهندسی دوآپس پرسیده می‌شوند. این کتاب را می‌توانید با مراجعه به سایت ناشر آن (انتشارات ناقوس) تهیه بفرمایید.



“

If you look at history, innovation doesn't come just from giving people incentives; it comes from creating environments where their ideas can connect.

”

Steven Johnson (born 1968), Science author & media theorist

code tale

شماره ۴

تهیه شده توسط: آرمان چم‌حیدری، امیرحسین صدر و آراس ولی‌زاده

هفته نامه code tale بهونه ای برای بیشتر دوندگی و نگاه همه جانبه به موضوعات رشته کامپیوتر هستش . دستیاران آموزشی درس برنامه سازی پیشرفته هدف والای خودشون رو بدین شکل می‌دونن که دانشجویان درس با مفاهیم روز رشته خود آشنا باشند و بتونن سریعتر علاقه خودشون رو پیدا کنن . امیدواریم از خوندن این قسمت لذت برده باشین .