ChatGPT

Detailed Command for Claude to Simulate Each Software Team Role

To ensure your final product is **comprehensive and high-quality**, you can give Claude a prompt that has it **simulate each key role in a software company** contributing to the project. The idea is that Claude will **“wear the hat” of each role** one by one – from leadership and product to engineering, design, and operations – and perform their part to polish and finish the project correctly. Below is a **detailed command** you can use, structured by functional areas, instructing Claude what to do in each role. Make sure to include the link to your project repository (e.g. the GitHub repo) so Claude has context on the software project in question.

Leadership & Direction Roles

- **CEO** – Ensure the project’s **vision and value** align with the company’s mission and market needs. As CEO, *verify that the final product meets the high-level goals* (e.g. user satisfaction, business model fit) and that it can be presented to stakeholders/investors as a success. Provide any high-level direction or scope adjustments if something is off-track with the original vision or company strategy.
- **CTO** – Review the **technical architecture and code quality** of the entire project. As CTO, *check that the technology choices and system design are scalable, maintainable, and secure*. Ensure best coding practices were followed (code readability, documentation, tests) and identify any technical debt that should be resolved. If any part of the stack is below standard, recommend improvements so the product is technically solid.
- **CPO (Head of Product)** – Evaluate the **product features and user experience** against the intended product vision. As CPO, *make sure the final product delivers real user value and has all the core features needed to solve the target problem*. Check that the **product roadmap** was fulfilled and prioritize any last-minute improvements that would significantly enhance user satisfaction. Ensure the product’s **user experience** aligns with market expectations and gather any user feedback for final tweaks.
- **COO** – Confirm that **operations and processes** are in place for a smooth launch and sustained delivery. As COO, *verify that the team’s workflows (deployment process, support process, incident response, etc.) are well-defined and efficient*. Ensure that any operational risks are addressed (for example, do we have a plan for maintenance updates or handling user growth?). The COO perspective should guarantee that the product can be delivered reliably to customers and that internal processes can support it post-launch.
- **CFO** – Check the **financial aspects** of the project’s completion. As CFO, *ensure the project stayed within budget and that any ongoing costs (servers, third-party services, etc.) are accounted for*. Verify that the pricing model or revenue mechanism of the product makes financial sense. If additional funding or budget adjustments are needed for launch (e.g. marketing spend, hiring support staff), highlight those. Essentially, confirm that the project is financially viable and prepared for any required financial reporting.

Product & Delivery Roles

- **Product Manager (PM)** – Take ownership of the **product outcome and backlog**. As PM, *review whether all the high-priority user stories and requirements have been implemented to meet the targeted outcomes*. Ensure that the **acceptance criteria** for each feature are met and that there are no outstanding critical items on the roadmap. If any user needs or project goals are unfulfilled, reprioritize or create a plan to address them. The PM should also confirm how success will be measured (KPIs/metrics) and prepare a plan for post-launch iterations based on user feedback.
- **Project/Program Manager** – Oversee the **timeline, scope, and inter-team coordination**. As the Project Manager, *verify that all project tasks have been completed on schedule and that any cross-team dependencies (front-end vs back-end integration, etc.) have been resolved*. Check the **project plan** to ensure nothing has fallen through the cracks for a proper launch (e.g. documentation, deployment steps). If any deliverables are missing or if any risks remain, coordinate with the respective roles to get them addressed promptly. Make sure a **launch checklist** is completed and that everyone knows their responsibilities on launch day.
- **Scrum Master / Agile Coach** – Ensure the **development process** is healthy and that the team is ready for release. As Scrum Master, *conduct a final review of the sprint artifacts: ensure that daily stand-ups, sprint reviews, and retrospectives have been yielding improvements and that any impediments have been removed*. Check the team's **velocity and morale**, ensuring no one is overburdened right before release. Facilitate a **blameless retrospective** for the project's development cycle, capturing lessons learned and making sure the team is continuously improving. Also, confirm that all Agile ceremonies (if using Scrum/Kanban) are wrapped up properly for this release cycle.

Engineering Roles

- **Frontend Engineer** – Finalize the **user interface and client-side functionality**. As the Frontend Engineer, *go through the UI thoroughly to fix any remaining bugs or visual inconsistencies*. Ensure that the app is **responsive** (works on different devices/screens), **accessible** (meets a11y standards), and optimized for performance (fast load times). If the project has a web interface, double-check cross-browser compatibility. Polish the UX details (animations, hover states, error messages) so that the interface feels smooth and professional.
- **Backend Engineer** – Finalize the **server-side logic, API endpoints, and database**. As the Backend Engineer, *ensure all APIs are functioning correctly and are secure (proper authentication/authorization, input validation to prevent injections, etc.)*. Optimize any slow database queries or memory-intensive operations for better performance. Verify data integrity and that all **business logic** is correctly implemented on the server. Also, check that the backend is properly documented (API docs for the front-end or any integration) and that the database migrations or schema are up-to-date for production.
- **Full-Stack Engineer** – Ensure **end-to-end integration** between frontend and backend is seamless. As a Full-Stack Engineer, *test the entire user flow from the UI to the database and back*. Identify any mismatches (e.g., the front-end expecting a field that the back-end doesn't provide) and fix them. Make sure that data flows correctly through the system and that the client and server handle errors gracefully. Essentially, verify that the **frontend-backend handshake** is smooth for all features – every button click or form submission triggers the right backend behavior, and the user gets the expected result without issues.
- **Mobile Engineer** – If the project includes a **mobile app**, finalize its implementation. As the Mobile Engineer, *ensure the iOS/Android app is feature-complete and polished according to platform guidelines*.

Fix any mobile-specific bugs and optimize performance (smooth scrolling, efficient network calls, proper caching). Ensure compatibility across different device models and OS versions. Verify that the app has appropriate **mobile UX** (touch targets, offline support if needed, push notifications if applicable) and that it cleanly integrates with the backend services. Prepare the app for release on app stores (correct icons, screenshots, metadata) if relevant.

- **Platform/Infrastructure Engineer** – Prepare the **infrastructure and developer tools** for launch. As a Platform/Infra Engineer, *ensure the continuous integration (CI) pipeline and deployment scripts are working without error, so that new code can be deployed easily*. Check infrastructure-as-code configurations (e.g., Docker, Kubernetes, AWS/Azure settings) to ensure consistency between environments (dev/staging/prod). Verify that the development and deployment **toolchain** (build scripts, package managers, etc.) is in good shape for maintainability. If the project requires scaling, ensure the infrastructure can be scaled up (e.g., load balancers configured, stateless services, database cluster ready).
- **DevOps / Site Reliability Engineer (SRE)** – Focus on **reliability, scalability, and monitoring**. As the DevOps/SRE, *make sure the application is ready for a stable production run*. Set up **monitoring and logging** for critical metrics (performance, errors, uptime) and establish alerts for any downtime or major issues. Conduct load testing or stress testing to ensure the system can handle expected user traffic, and address any performance bottlenecks uncovered. Verify backup and recovery processes (for databases or critical services) are in place. Essentially, ensure that the deployment will be **smooth and the system reliable** under real-world conditions, and prepare an **incident response plan** in case something goes wrong in production.
- **Security Engineer** – Perform a thorough **security audit** of the project. As the Security Engineer, *review the code and configurations for vulnerabilities*. Check for common security issues (SQL injection, XSS, CSRF, etc.), ensure proper encryption is used for data in transit and at rest, and verify that authentication and authorization mechanisms are robust. If the project deals with user data, confirm compliance with privacy standards and that sensitive data (like passwords, API keys) is handled safely (e.g., hashed, not exposed in logs). Run penetration tests or use security scanning tools on the application. Address any identified security flaws so that the product is **secure-by-default** before release.
- **QA / SDET (Quality Assurance / Software Development Engineer in Test)** – **Test the product thoroughly** to uphold quality. As QA or SDET, *execute all relevant test cases (unit tests, integration tests, end-to-end tests, exploratory manual testing) to ensure every feature works as intended*. Verify that **edge cases** and error conditions are handled gracefully without crashes. Ensure **regression testing** is done (nothing new broke existing functionality) and that the **test coverage** is adequate. It's your job to **catch any bugs** or UX issues before real users do. In particular, make sure **every release is thoroughly vetted** and meets the quality standards before it reaches users [1]. If any defects are found, communicate with engineers to get them fixed and re-test the fixes.
- **Data Engineer** – Take care of any **data pipelines or databases** involved. As the Data Engineer, *ensure that data flows (ETL/ELT processes) are correctly set up and schedule/runtimes are reliable*. If the project involves data storage or a data warehouse, verify that the schemas are well-designed and the data is being correctly populated. Optimize database performance (adding indexes or caching as needed) to handle production load. Also, confirm that data backup routines are in place, and that any data privacy or retention policies are being followed. If the product will scale to large data volumes, ensure the architecture (e.g., using big data tools or cloud services) can accommodate future growth.
- **ML/AI Engineer** – If the project includes **machine learning or AI features**, finalize those components. As the ML/AI Engineer, *ensure that any ML models are well-trained on appropriate data and integrated properly into the app*. Validate the model's performance (accuracy, bias, etc.) using the

latest data – if it’s not satisfactory, consider a last round of tuning or retraining. Check that the inference pipeline is efficient (for example, the model doesn’t slow down the app and can scale under load). Also, implement fallback behavior if the model fails (e.g. an error message or default recommendation so the user isn’t stuck). Verify that using the model complies with ethical guidelines and that you’re not inadvertently introducing bias or privacy issues. Document the ML feature for future maintainers (how the model was trained, how to retrain, etc.).

- **Release Engineer** – Plan and execute a **smooth release** of the final product version. As the Release Engineer, *coordinate the release schedule and ensure version control is properly managed (e.g., tagging the release in Git)*. Double-check that all release artifacts (binaries, install packages, container images, etc.) are built correctly and reproducibly. If using feature flags or a phased rollout, prepare those configurations. Write or review the **release notes** to document new features, bug fixes, and any known issues. Finally, deploy the software to the production environment (or app store, etc.), following the predetermined rollout strategy (perhaps a canary release or beta period first) to catch any issues early. The goal is a **painless deployment** where end-users can start using the product without hiccups.

Design & Research Roles

- **Product/UX Designer** – Conduct a final **design review** of the user interface and experience. As the Product Designer, *ensure the UI is intuitive, visually appealing, and consistent*. Go through each screen or page of the application to confirm that it follows the design specifications and brand style guides (correct colors, typography, spacing, etc.). Fix any small UI/UX issues (alignment problems, inconsistent icons, etc.) that might have been introduced during development. Verify that the **user flow** is smooth – for example, the navigation is clear, and users won’t get lost or confused using the product. If any part of the experience is suboptimal, propose quick design improvements.
- **UX Researcher** – Validate the product with a **user perspective**. As the UX Researcher, *if possible, perform a last-minute round of user testing or review any feedback from beta users*. Ensure that the product’s design truly solves the users’ problems and is easy to use in real-world scenarios. Identify any **usability issues** or areas of confusion for users (even minor annoyances) and report them to the team. Given the late stage, also prepare a plan for post-launch user research (like gathering user feedback, analytics on feature usage) to drive the next iteration. The key is to make sure the team isn’t launching with any known major UX problems unaddressed.
- **Design Systems Engineer** – Ensure **design consistency and standards** are upheld. As the Design Systems specialist, *check that the implemented UI components conform to our design system or style guide*. Verify that things like buttons, form fields, modals, etc., are used consistently across the app and match the approved components in terms of look and behavior. Ensure **accessibility standards** are met: for instance, color contrast should be sufficient for readability, and UI controls should be usable via keyboard and screen readers. If any new UI patterns were created during development, document them and consider adding them to the design system for future reuse. Essentially, certify that the final product’s design is not only good-looking but also **consistent and accessible**.
- **Content Designer / UX Writer** – Polish all **user-facing copy and content**. As the UX Writer, *review every piece of text in the application – labels, tooltips, error messages, notifications, dialog content, onboarding instructions – to ensure clarity and the right tone of voice*. Make sure the language is **concise and helpful**; users should easily understand what each message or prompt means. Check for consistency in terminology (e.g., don’t call the same concept by two different names in different places). If the product has any user documentation or FAQ, update those to reflect the final state of

the software. Good content design will eliminate user confusion and make the product feel polished and professional.

Data & Insights Roles

- **Data/BI Analyst** – Set up **analytics and metrics tracking** for the product. As the Data Analyst, *ensure that key user actions and product metrics are being tracked (through analytics tools or logging) so that the team can measure the product's performance after launch*. For example, if the product is a web app, confirm that tools like Google Analytics or other telemetry are integrated for page views, sign-ups, conversions, etc., as appropriate. Define the **Key Performance Indicators (KPIs)** that align with the product goals (e.g., daily active users, retention rate, task success rate) and make sure the means to collect those metrics are in place. You might also prepare initial **dashboard reports** or SQL queries so the team can easily monitor these metrics post-launch. After release, this will help in making data-driven decisions.
- **Data Scientist** – If applicable, plan for **experimentation and advanced analysis**. As the Data Scientist, *ensure that the team can learn from the launch and continue improving the product*. Design any A/B tests or **experiments** that will be run at or after launch (for example, testing two variants of a feature to see which performs better) and make sure the framework to conduct those tests is ready in the product. If the product involves any algorithms or data analysis internally (e.g., a recommendation system, user behavior analysis), double-check those components for accuracy and bias. Also, consider creating a **baseline analysis** of any pre-launch data (if available from beta tests or simulations) to have something to compare against after launch. The Data Scientist's role at this stage is to set up the product for **continuous learning and optimization** once real user data starts coming in.

Go-to-Market Roles

- **Product Marketing Manager (PMM)** – Craft the **messaging and launch strategy**. As the PMM, *prepare all user-facing communications about the product – this includes the value proposition, feature highlight descriptions, app store descriptions, website copy, press releases or announcement blog posts*. Ensure that the **positioning** of the product is clear and compelling: it should be obvious what problem the product solves and why it's unique. Coordinate the **launch plan** (date and time of release, any launch event or campaign) and make sure sales or customer-facing teams are trained on the product's benefits and talking points. The PMM should also gather marketing materials (screenshots, demo videos, etc.) and be ready to publicize the launch to drive user adoption.
- **Growth Marketing** – Plan and implement **user acquisition and retention tactics**. As the Growth Marketer, *ensure that channels for bringing in users are set up and optimized*. For example, verify that SEO is handled (the product website has relevant keywords, meta tags, and content to rank well in search results) and that any paid advertising campaigns (Google Ads, social media ads) are ready to launch. Set up **user onboarding emails or tutorials** to help new users get started, and perhaps **referral programs or promotions** to encourage sign-ups. Also, prepare to measure conversion funnels – understand where potential users might drop off – and have plans to address those (maybe through email reminders or in-app nudges). The goal is to not just launch and hope for the best, but actively work to **attract and retain users** from day one.
- **Community/Developer Relations (DevRel)** – Engage with the **user or developer community**. As the DevRel or Community Manager, *set up support forums, chat groups, or other community channels where early users can ask questions and share feedback*. If the project is developer-facing or has an API,

ensure that **developer documentation and examples** are ready, so that third-party developers can easily integrate or extend the product. Consider writing tutorial blogs or hosting a webinar to walk through the product for interested users or developers. Also, be prepared to act as the liaison between the community and the development team – gather common questions or issues from users and relay them to the team for quick resolution. Having a strong community presence at launch helps in building user trust and loyalty.

Sales & Customer Roles

- **Sales Development Representative (SDR) / Business Development Rep (BDR)** – (If your project has a **sales component** such as B2B clients) Make sure **lead generation** processes are ready. As an SDR/BDR, *compile a list of potential leads or target customers and ensure we have outreach sequences or campaigns prepared to contact them*. Create or refine the sales pitch for initial conversations, focusing on how the final product addresses client needs. Essentially, be ready to drum up interest in the product by reaching out to prospective customers or partners as soon as it launches.
- **Account Executive (AE)** – Be prepared for **product demos and closing deals** with interested customers. As an AE, *make sure you have a compelling demo of the final product and that all pricing/licensing info is finalized*. Work closely with the PMM and SDR to understand the product's value proposition deeply, so you can effectively communicate it. If any early adopters or beta customers exist, gather their testimonials or case studies to bolster sales conversations. The AE's goal is to convert the product's launch buzz into actual signed customers, so ensure contracts, proposals, and the CRM are all set for new customer onboarding.
- **Sales Engineer** – Handle any **technical deep-dives or proofs-of-concept** for potential clients. As the Sales Engineer, *ensure that you can support the AE by answering in-depth technical questions about the product*. If enterprise clients need a custom integration or specific security info, prepare those materials (for example, have a demo environment or sandbox ready, prepare architecture diagrams or compliance documents). Test the process of deploying or integrating the product in a customer's environment if that's applicable, so you know it works smoothly. Essentially, be the technical backbone of sales to show that the product is robust and can meet complex requirements if needed.
- **Customer Success Manager (CSM)** – Set up the **post-sale customer onboarding and support** process. As the CSM, *create an onboarding checklist or guide for new customers to get started with the product successfully*. Schedule any training sessions or Q&A calls for early customers, and ensure you have a way to track customer satisfaction and product usage (so you can proactively reach out if a customer seems to be struggling). The CSM should also prepare a feedback mechanism: as customers start using the product, collect their feedback and issues to relay back to the team for continuous improvement. The goal is to turn early customers into happy **long-term advocates** by making sure they achieve their desired outcomes with the product.
- **Support / Tech Support** – Ensure **customer support channels** are live and ready. As the Support lead, *set up the helpdesk or ticketing system (for example, Zendesk or a shared support email) and draft answers/knowledge-base articles for frequently asked questions*. Make sure there's a process to handle incoming issues: triage bugs vs. usage questions, and route issues to engineers if needed. Test the support process by simulating a few common issues and seeing if the team can resolve them quickly. Also, prepare any on-call rotation if this is a 24/7 service. Being ready to support users from day one will greatly improve user confidence and catch any problems early.
- **Professional Services / Implementation** – If the product requires **custom setups or enterprise integration**, have a plan for that. As the Professional Services team, *ensure that for any client that needs a tailored solution or custom configuration, you have the resources and scripts to do so*. This could

include data migration from an older system into your product, custom feature toggling, or integration with the client's other software. Write up any **integration guides or implementation plans** for complex setups. Even if not needed at launch, having this ready will enable you to serve those valuable big customers who might ask for pilots or custom work. It's all about making sure no customer need falls through the cracks just because it's beyond the out-of-the-box features.

Operations & Company Backbone Roles

- **People Operations / HR & Recruiting** – Though not directly tied to the product features, ensure the **team is supported and growing healthily**. As the HR/People Ops role, *check in on the team's well-being after a hard push to finish the project*. Make sure there are plans to celebrate the launch (to boost morale) and possibly that any overtime or stress is addressed. If the next phase requires hiring (maybe more developers for new features or support staff), have job postings and recruiting pipeline ready. A happy and well-organized team will continue to deliver great products, so this role safeguards the human side of the project's success.
- **Finance & Accounting** – Take care of any **financial/logistical wrap-ups** for the project. As Finance, *ensure all bills are paid (cloud hosting, domain registrations, software licenses, etc.) and that the project's spending is tracked against the budget*. Prepare any financial reports needed if this project is tied to a larger program or if investors need an update on how funds were used. If the product will generate revenue (sales or subscriptions), confirm that payment processing is set up and tested, and that there's a plan to record revenue properly. Essentially, tidy up the financial side so there are no monetary surprises at or after launch.
- **Legal & Privacy** – Verify the **legal compliance** of the final product. As the Legal advisor, *ensure that all user terms of service and privacy policies are up to date and cover the features of the product*. Check that the product doesn't violate any licenses (for example, all third-party libraries are used in compliance with their licenses) and that any open-source attributions are done. If the product involves user data, ensure **privacy regulations** like GDPR or CCPA are complied with – e.g., users have consent prompts if needed, and there's a way to handle data deletion requests. Also, ensure any required **contracts** (like with a client or vendor) are signed and in place for launch. Covering these legal bases protects the company and the product in the long run.
- **IT / Workplace** – Handle any **internal IT preparations**. As IT support, *ensure that the team has all tools working for launch day – for example, everyone has access to the production systems, the domain DNS is configured correctly, SSL certificates are in place, and any internal dashboards or admin tools needed for the product are accessible*. Also, double-check things like company accounts on app stores or cloud platforms are using the correct credentials and are secured. Essentially, make sure the "office" side of tech is not a blocker: all equipment, accounts, and networks should be ready and secure for the product's operation.
- **Compliance / Data Protection / Trust & Safety** – Guarantee the **product meets any industry regulations and ethical standards**. As the Compliance or Trust officer, *ensure any necessary certifications (PCI DSS if dealing with payments, HIPAA if health data, etc.) are addressed or on the roadmap*. Conduct a final review for any **compliance checklist** relevant to your domain (security audits, penetration testing results, data processing agreements). Also, if the product allows user-generated content or interactions, have a **content moderation or abuse prevention plan** in place – for example, filters for offensive content or a way for users to report issues. Make sure user data is handled with care (encryption, limited access) to build trust. By covering compliance and safety, you reduce the risk of legal issues or user trust issues after launch.

Final instruction in the prompt: After going through **each role above**, Claude should compile the findings and actions from all perspectives and **apply them to the project repository** (e.g., `[valuerpro-project](https://github.com/Malith-nethsiri/valuerpro-project)`). This means for each role's insight, Claude should either make the necessary changes in the project (if possible via code or configuration tweaks provided in the repo) or at least output a clear list of what needs to be done to address that area. By sequentially simulating all these roles, we ensure that *everyone's expertise intertwines to build an exceptional final product* [2], covering all angles from vision and usability to code quality and market readiness. The end result should be a product that is **polished, robust, user-friendly, and ready for a successful launch**, just as it would be if a full software company's team had collaborated to finish it.

---

[1] [2] 11 Key Roles in a Software Development Team [+3 Emerging]
https://alcor-bpo.com/10-key-roles-in-a-software-development-team-who-is-responsible-for-what/