Business considerations

# Advanced concepts

Abrupt impedance changes at the boundary of a communication channel cause inefficient energy transfers. This low efficiency of information transfer can be improved by adding a series of incremental, smaller changes between the two original ones.

# IDEAL CUSTOMER PERSONA

## B2B

We’re looking for a small to medium US based software company with 30 to 100 engineers, and enough profit margin to not be too worried about the 1000 USD / month license for Code Awareness (for 30 users). They would adopt Code Awareness with limited scope, perhaps a single product with a sub-set of their engineering teams.

The company culture is innovative and at least one internal team is willing to try new workflows and tools, and understands the risks involved. Ideally, the company feels the pain of merging code every day, and would be eager to try a new method.

The company is currently using GitHub, GitLab or BitBucket. Their Pull Requests take a long time and involves many engineers gathered in a meeting for weekly code review. <— THIS is where Code Awareness shines, currently. Code Reviews once a week are very distracting and usually not very effective due to engineers focusing on their own tasks for an entire week, and then suddenly being thrown into other people’s code.

Another ideal opportunity could present itself in the form of partnership: the customer recognizes the value of Code Awareness and offers to contribute strategically.

The current pricing on Code Awareness site is setup with only the limited, existing functionality in mind. Multiple products / tools will be added in the future, some of which will be priced independently (like AI assistants).

There are no specific laws or regulations or certifications required to use Code Awareness, and we ensure the privacy and protection of intellectual property (the customer codebase and communications).

## B2C

We’re looking for individuals world-wide, who have a side project, or contribute to open source projects. Ideally they are frequent contributors or reviewers to a major open source project.

The customer communicates in any language, but English is preferred.

The customer is not afraid to install software on their computer. Alternatively, they are comfortable analyzing the Code Awareness source code to judge for themselves the risks of such installation.

The customer loves Star Trek and its culture.

The customer works primarily on projects with external contributors, not a personal project where they are the only contributor.

The customer feels the pain and chaos of maintaining a repository with a large number of contributors.

Ideally the customer has some personal research or is interested in the fields of sociology and management.

The customer has a good enough financial status to pay 15 dollars /month for Code Awareness. Note that Code Awareness can be used for free for open source repositories, but a paid license offers extra benefits (to be defined at a later date).

## Marketing Strategy

* Hacktoberfest
* Post on other famous projects "we're using your library". E.g. Changesets

## Pricing Strategy

Option 1: Tiered Pricing based on Team Size

- Small Team (Up to 10 users): $X per user per month

- Medium Team (11-50 users): $Y per user per month

- Large Team (51+ users): Custom pricing based on negotiation

Option 2: Usage-based Pricing

- Base Fee: $Z per month (includes a certain amount of monthly traffic, e.g., 500MB)

- Additional Traffic: $A per GB (beyond the included monthly traffic limit)

Note: The above pricing options can be combined to offer flexibility and cater to different customer segments. For example, you can offer tiered pricing based on team size and add an additional fee for excessive usage beyond the included traffic limit.

Factors to Consider:

1. Market Research: Evaluate the pricing of similar collaboration tools in the market to ensure competitiveness and perceived value.

2. Customer Segmentation: Consider the size and type of customers (startups, enterprises, etc.) to determine appropriate tiered pricing.

3. Value-based Pricing: Assess the value CodeAwareness brings to teams in terms of improved collaboration, code quality, and productivity to justify pricing.

4. Cost Analysis: Consider development and maintenance costs, infrastructure expenses, customer support, and potential future enhancements.

5. Scalability: Keep in mind the scalability of pricing as the product gains popularity and attracts larger teams.

Communicate the Pricing:

- Clearly communicate the pricing structure and the value proposition to potential customers during the sales pitch and marketing materials.

- Provide a transparent breakdown of the features and benefits included in each pricing tier to help customers make informed decisions.

- Offer a free trial or freemium version with limited features to allow potential customers to experience the value of CodeAwareness before committing to a paid plan.

It's important to note that the pricing mentioned above is just a starting point. Conducting market research, analyzing customer feedback, and monitoring the competitive landscape will help refine the pricing strategy over time to ensure it remains competitive and profitable.

**Advantages of Working with CodeAwareness**

1. Real-Time Awareness: CodeAwareness provides real-time code highlighting, allowing team members to be constantly aware of each other's code changes. This eliminates the need for manual coordination and reduces the chances of conflicts and merge issues.

2. Enhanced Collaboration: By promoting constant awareness of code changes, CodeAwareness fosters a collaborative environment. Team members can easily identify dependencies and potential conflicts, leading to smoother collaboration and improved communication.

3. Streamlined Code Reviews: CodeAwareness simplifies code review processes by providing a centralized list of all files modified by the team. This makes it easier for reviewers to track changes, provide feedback, and ensure high-quality code.

4. Time Savings: With CodeAwareness, developers can save time by avoiding unnecessary context switching. They can focus on their own coding tasks while being aware of ongoing code changes without actively monitoring git branches or merge requests. In addition, one developer could easily hand over their work to another, something that is usually difficult in today's software engineering.

5. Skill Enhancement: By making code changes more visible and accessible, CodeAwareness promotes knowledge sharing and learning within the team. Engineers can gain insights into different coding approaches, techniques, and problem-solving methods, leading to skill enhancement and professional growth.

## Disadvantages of Working with CodeAwareness

1. Learning Curve: Introducing a new tool like CodeAwareness may require some initial training and adaptation from team members. They will need to familiarize themselves with the tool's features and incorporate it into their existing workflow.

2. Dependency on the Tool: CodeAwareness becomes an integral part of the team's workflow, and any interruptions or technical issues with the tool may temporarily affect productivity until the issues are resolved.

3. Integration Challenges: CodeAwareness needs to seamlessly integrate with existing development environments, such as IDEs and text editors. Compatibility issues or difficulties in integration may arise, requiring additional effort to set up and configure the tool.

4. Cost Considerations: Implementing CodeAwareness may involve additional costs, depending on the pricing structure chosen. Organizations need to evaluate the value proposition of the tool against the associated expenses to ensure a positive return on investment.

5. Potential Overhead: While CodeAwareness aims to enhance collaboration, there is a possibility that excessive notifications and code highlighting could cause distractions or overload team members with unnecessary information. Proper customization and configuration are necessary to strike a balance between awareness and individual focus. However, this can be avoided by closing the CodeAwareness panel and working in Zen mode.

6. Code Ownership: With real-time code highlighting, team members might feel more protective of their code and concerned about others making direct modifications without prior communication. It's important to establish clear guidelines for collaboration and foster a culture of open communication to address this potential tension.

It's important to consider that the advantages and disadvantages may vary based on the specific needs and preferences of each development team. Conducting a thorough evaluation and obtaining feedback from the team members themselves can provide more accurate insights into how CodeAwareness would impact their workflow.

## Imagine Code Awareness

* functions without side effects are automatically added to the main branch
* unused functions are highlighted; CodeAwareness provides a list of such functions across all repos.
* Highlight modes: diffs against current HEAD, against cSha, lines commented, security aspect, data aspect, cache aspect, etc
* record the first contributor to use a certain pattern in a repo
* i went through several iterations of the project, starting in Tokyo
* this project promise is...
* we will communicate with you every week

## Imagine - What If

* what if you could become aware of code changes made by your peers?
* what if you could merge individual pieces of code without even going through a merge request?
* just imagine where you could go from here?