



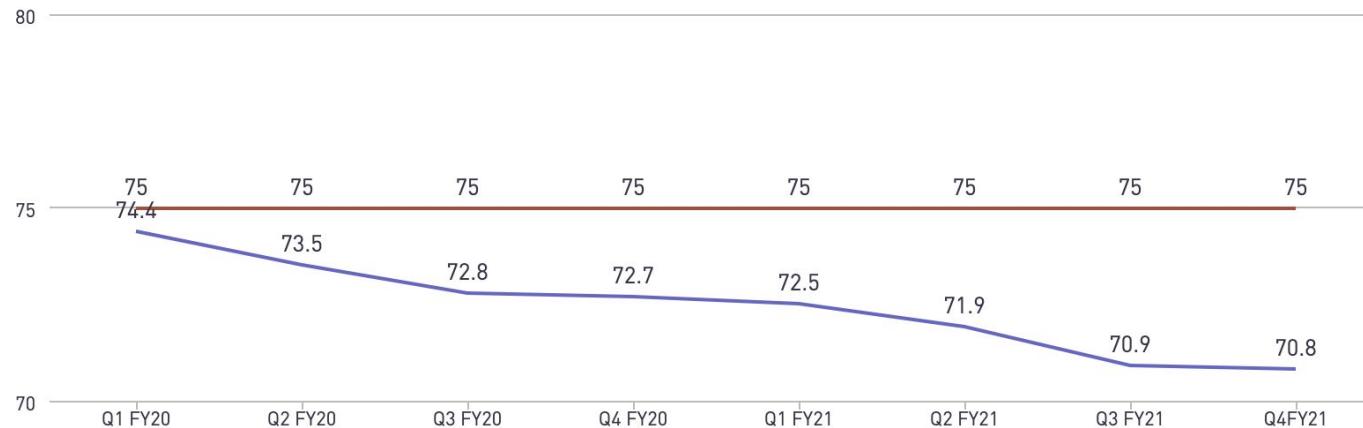
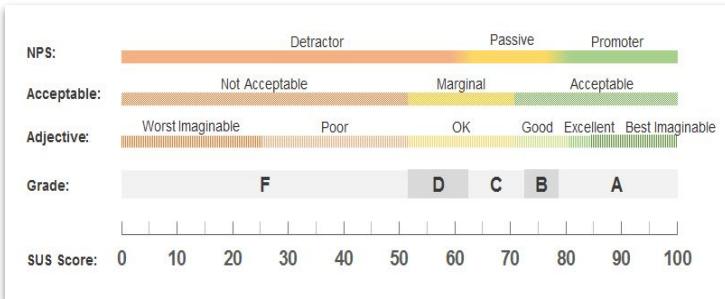
System Usability Scale
FY21-Q4

Perceived system usability over time



The overall System Usability Scale Score for FY21-Q4 is **70.8**, with a margin of error is +/- 1.3 at the 95% confidence level. The survey received 513 responses.

GitLab's perceived usability rates as a C and has steadily declined since we first started running the survey.



SUS score by cohort



SUS scores are broken down by cohorts to determine whether perception of usability differs between segments of users. We consider the differences “**meaningful**” if they are outside of the margin of error. Some insights around the cohorts:

- The cohort scores don’t dramatically differ from the overall SUS score.
- The **Free** cohort score is meaningfully different from both the **Paid** cohort and the overall SUS score.
- The **Free** score is a C+ and considered “Good” while the **Paid** score is a C and considered “OK.” We saw this difference in FY21-Q3 as well.
- We do not see a meaningful difference between **New** and **Mature** cohort scores.

	Q4 FY2021	Diff. from last quarter
Paid users	69.7	+0.5
<i>Margin of error</i>	+/- 1.8	
<i>Number of responses</i>	306	
Free users	72.4	-0.0
<i>Margin of error</i>	+/- 1.9	
<i>Number of responses</i>	207	
New users (<180 days)	71.4	+1.6
<i>Margin of error</i>	+/- 1.8	
<i>Responses</i>	215	
Mature users	70.6	-0.3
<i>Margin of error</i>	+/- 1.9	
<i>Number of responses</i>	284	

Top usability themes that impact our SUS score



We analyze user feedback to uncover themes and targeted sub-themes that impact usability and lower the SUS score. **These tend to span across the product and may not be directly related to a specific stage.**

We consistently see these **6 top themes** every quarter.

Visibility of system status

- Let users know what's happening

System performance

- Lack of real-time updates
- We're not remembering information
- When the system is slow
- General weirdness

Navigation & Discoverability

- It's difficult moving around GitLab
- It isn't easy to find what you need, when you need it
- We can do more to help our users along

General usability

- Make it easier
- Reduce confusion
- Use standard terms

Search

- Not meeting users' expectations

Merge Requests

- Improve the MR experience

Theme
↳

Sub-themes
↳

Additional themes impacting our SUS score



In addition to the top themes, the following themes were often tied to responses from users who gave us lower scores. We have seen these in past SUS verbatims, too.

Documentation/help	Managing groups and projects	Missing features/functionality
<p>Defined as: Providing guidance, tutorials, and helpful tips to users as they complete tasks in GitLab. Making it easy to find clear examples and use cases in our documentation.</p>	<p>Defined as: Making it easier for organizations to manage work, settings, and members across groups and projects.</p>	<p>Defined as: Improving the quality of life of our users by enhancing existing features, fixing gaps in usability, and optimizing workflows.</p>



Some early sub-themes are starting to emerge, as we continue to collect more data:

Include more examples and use cases in the documentation

- “When reading objects with the API, rather than merely showing sample output, give a definition of what each property in the output means...”
- “Some quick video tutorials might help beginners not to feel too overwhelmed by all the settings to set up when starting.”
- “With the pipelines docs and integration you could go a lot further. Adding more completes pipeline examples will be a lot useful, and same things on how to use Gitlab runner will be great.”

Improve documentation of CI/CD features

- “There are a lot of conflicting functionalities in the syntax and it's not obvious which one to use... It adds a lot of cognitive burden for newcomers when we want to build our pipelines...”
- “The point I see lacking is documentation around using yaml and/or building full scale projects. There is some ruby code and rails code but that does not help me when trying to use bash, asp.net and python for testing some use cases...”

Managing groups and projects - early sub-themes



Make it easier to manage memberships

- “We have 3 top level groups and many groups and projects nested underneath...making sure we have removed a user from all of the groups and projects in which they could potentially exist is cumbersome.”
- “[It] was a little confusing to find how to transfer the ownership of a project from me to another user. I basically had to create a group, transfer the project to that group, add the user to that group and have that user take ownership of the project.”

Help users differentiate between project- and group-level features

- “During sprint planning, we pull up the group issue board. I click on an issue. I am taken to the *project* issue board, which looks so much the same that I assume that clicking on the project menu to the left on boards will take me back to the group board...”
- “Overall having the same interface for "Groups" and "Projects" is misleading and confusing. A group might be composed of projects, but a group *is not a* project.”

Make project- and group-level features and configurations more consistent

- “Lots of stuff is only configurable on projects, and not on groups, which becomes very cumbersome once we have a non trivial number of projects that need similar setup.”
- “[There is] no source control for the group wiki. We've just designated one repository's wiki as the 'official cross repo/project wiki.'”



Help users save steps

- “[It’s] frustrating that there’s not a [Pipeline] feature (that I’m aware of) that says ‘once stage X is done and completed successfully, go on to start stage Y automatically.’”
- “It would be really great if Gitlab offered more actions to use in issues/mr. I feel like the actions are limited and you have to use the UI in many use cases.
- “The code quality integration is pretty cool, but it would be great if we could sort through the existing thousands of issues (from before we made it part of our process) from within Gitlab so we can create issues out of them.”

Fix gaps in usability

- “Gitlab has a lot of features at various states of completeness sometimes missing features you would expect. Some of the more powerful features (ex gitlab-ci) take time to master.”
- “Sometimes we have more issues than we can reasonably fit into a milestone and there is no good way to prioritize them to see what we can get in...so relative ordering that actually sticks to the issues (instead of e.g. new issues cutting the line) would be very helpful.”



Diving deeper into the SUS score

A closer look at the System Usability Scale questions



We calculate scores for the individual questions to help us gain additional insight into specific aspects of the user experience.

Process: We take the normalized single question score and multiply it by 25, to align with the scale used for the overall SUS score.

The overall SUS score is the average of all the individual scores.

Interpretation: Individual scores that are lower than the **overall SUS score (70.8)** indicate areas of concern. We use SUS themes to dig deeper into these questions and uncover usability issues.

Question	Score
I think that I would like to use GitLab frequently.	83.3
I found GitLab unnecessarily complex.	65.4
I thought GitLab was easy to use.	71.1
I think that I would need the support of a technical person to be able to use GitLab.	74.4
I found the various functions in GitLab were well integrated.	72.2
I thought there was too much inconsistency in GitLab.	70.8
I would imagine that most people would learn to use GitLab very quickly.	66.5
I found GitLab very cumbersome to use.	67.0
I felt very confident using GitLab.	74.4
I needed to learn a lot of things before I could get going with GitLab.	62.8



Users are encountering complex and cumbersome experiences

Looking closer at the detailed questions for clues, we found that scores are low for the questions related to the complexity of GitLab.

This is most likely related to the **General Usability** theme where both small and large pain points can negatively impact the overall experience. Confusing, unintuitive, and clunky experiences can add up and significantly impact users' ability to use GitLab.

SUS themes such as **Navigation and discoverability**, **Managing groups and projects**, **Search**, and **MR Experience** were also tied to feedback from users who gave us low ratings for these questions.

Question	Score
I think that I would like to use GitLab frequently.	83.3
I found GitLab unnecessarily complex.	65.4
I thought GitLab was easy to use.	71.1
I think that I would need the support of a technical person to be able to use GitLab.	74.4
I found the various functions in GitLab were well integrated.	72.2
I thought there was too much inconsistency in GitLab.	70.8
I would imagine that most people would learn to use GitLab very quickly.	66.5
I found GitLab very cumbersome to use.	67.0
I felt very confident using GitLab.	74.4
I needed to learn a lot of things before I could get going with GitLab.	62.8

How can we reduce complexity in GitLab?



Some examples of how we can reduce complexity in GitLab:

- Help users complete tasks without context-switching
- Make it easier to find where you were last working in GitLab
- Reduce visual clutter in the interface
- Help users differentiate between groups and projects by providing clear visual indicators
- Improve discoverability of functionalities and features



Learning GitLab can be difficult

Scores were also low for both questions that related to learning how to use GitLab.

Learnability is a component of usability that considers how easy it is to accomplish tasks on the first and subsequent time. Users need clear communication and guidance throughout GitLab to help them complete their work and recover from errors.

SUS themes such as **General Usability**, **Documentation/help**, and **Navigation and discoverability** were also tied to feedback from users who gave us low ratings for these questions.

Question	Score
I think that I would like to use GitLab frequently.	83.3
I found GitLab unnecessarily complex.	65.4
I thought GitLab was easy to use.	71.1
I think that I would need the support of a technical person to be able to use GitLab.	74.4
I found the various functions in GitLab were well integrated.	72.2
I thought there was too much inconsistency in GitLab.	70.8
I would imagine that most people would learn to use GitLab very quickly.	66.5
I found GitLab very cumbersome to use.	67.0
I felt very confident using GitLab.	74.4
I needed to learn a lot of things before I could get going with GitLab.	62.8

How can we help users learn to use GitLab?



Some examples of how we can help users learn how to use GitLab:

- Provide more guidance and in-context help in the UI
- Provide more examples and tutorials in the documentation
- Make it easier to locate commonly needed information and features
- Improve and simplify setup processes for features such as Runners and AutoDevOps

What we're doing now



Our Product and UX departments have jointly agreed to have a shared OKR every quarter to focus on usability. Every quarter, we will choose to focus on one of the top 6 SUS themes.

As part of the [shared FY22-Q1 OKR](#), we will implement meaningful UX improvements that align with our **UI Polish** and **System speed & reliability** themes.

Focus by product section:

- **Dev:** UI Polish
- **Ops:** UI Polish
- **Secure:** System speed & reliability
- **Enablement:** System speed & reliability



Appendix

What is the System Usability Scale?



The System Usability Scale (SUS) is a standardized, 10-item questionnaire that measures the ease of use of websites, software, mobile devices, and other technological applications.

We conduct the SUS survey every quarter as a way to track the perceived usability of GitLab over time.

I think that I would like to use GitLab frequently.

I found GitLab unnecessarily complex.

I thought GitLab was easy to use.

I think that I would need the support of a technical person to be able to use GitLab.

I found the various functions in GitLab were well integrated.

I thought there was too much inconsistency in GitLab.

I would imagine that most people would learn to use GitLab very quickly.

I found GitLab very cumbersome to use.

I felt very confident using GitLab.

I needed to learn a lot of things before I could get going with GitLab.

How we run the survey



We send the SUS survey to a randomized selection of GitLab.com (SaaS) users. These users must have a minimum threshold of 10 product events across at least 2 stages in the previous 30 days.

We track the following cohorts over time:

- **Paid users:** Users associated with a paid subscription (whether purchased or gifted by GitLab).
- **Free users:** Users that are not currently associated with any paid subscriptions and are using the Free plan.
- **Mature users:** Users that have a tenure of 180 days or more.
- **New users:** Users with a tenure of less than 180 days.

All cohorts have a minimum sample size of 200 users.

Resources



FY21-Q4 results spreadsheet:

<https://docs.google.com/spreadsheets/d/1mJdGSs1TeBdb-ol38I4leebEgh5R69xmRIPYIrPky9Y/edit?usp=sharing>

A deeper look at SUS themes (slide deck):

https://docs.google.com/presentation/d/1kMZcRnOjuQ0K3TtT1YxehhFb5-Ej_cr2mXxeeJ9sbZ0/edit?usp=sharing

System Usability Scale video series (July 2020):

https://www.youtube.com/watch?v=OIF5S6TZdl8&list=PL05JrBw4t0KrPNM6WFtrsVM8tWwkYDyio&index=1&ab_channel=GitLabUnfiltered

System Usability Scale approach and methodology:

<https://about.gitlab.com/handbook/engineering/ux/performance-indicators/system-usability-scale/>



GitLab

Thank you!