

P4000R0:

To TS or not to TS: that is the question

Whether 'tis nobler in the mind to suffer
The slings and arrows of outrageous fortune,
Or to take arms against a sea of troubles,
And by opposing end them.

Date: 2024-05-22 Programming Language C++ Audience: All
WG21

H. Hinnant, R. Orr, B. Stroustrup, D. Vandevoorde, M. Wong

The role of TSes

We (DG) recommend

- Answer the key question:
 - WHAT ARE we hoping to LEARN through a TS must be clearly specified.
 - WHAT ARE the exit criteria of the TS to IS must be clearly specified.
- Use TSs for library components.
- Don't use TSs for a language feature unless the feature is a mostly self-contained unit.
- Don't use a TS simply to delay; it doesn't simplify later decision making. Have concrete and articulated criteria for completion.

TS vs IS: Possible question a TS should answer

- Is there an implementation?
- Is it a Library or Language proposal, or both?
- Is the proposal a foundational proposal, meaning many other C++ aspects/proposals depend on it, and/or does it depend on many other C++ aspects/proposals?
- Is it independent on aspects of the language?
- Are there competing design proposals?
- Is the proposal so complicated or large to fear there will be error in design decisions?
- Is it a research idea?
- Is there substantial invention?
- Can it be staged?
- Is there a subpart that deserves to be in IS?
- Is the wording complicated or unconventional?
- Is applying to library important for the feature?
- Will the proposal benefit from early integration (can it be applied to a WP)?
- Will you get feedback/testing only after TS publication or IS publication?
- Is there a motivation to slow down a proposal?
- Explicitly state the acceptance criteria for the TS into IS.
- Are you juggling a large number of related or dependent proposals (other proposals that depend on this proposal)?
- Are you aiming for user feedback?
- Are you aiming for implementation feedback?
- Is there a scheduling concern to make C++xx for it or its dependents?

Summary To TS or not to TS

- WGs SGs decide on TS or IS route and write proposal supporting direction
- The key question (repeated here because it is so important):
 - WHAT ARE we hoping to LEARN through a TS must be clearly specified.
 - WHAT ARE the exit criteria of the TS to IS must be clearly specified.
- All the previous (page) questions and their answers should be considered
- The cost of a Language TS is almost that of an IS due to tightly coupled interaction
 - TS costs Resources, people, time, build/compile system
 - Delay of a TS likely costs at least an extra cycle
 - Library TS have been more useful
 - Ask is it better to just continue working towards an IS in the next release instead of adding a TS into the cycle
 - Should it just stabilize without needing a TS churn to get feedback
- We urge SGs to explicitly poll for this and their supporting reasons
- DG will offer **non-binding** advise in some cases as whether TS or IS route is preferred, or have you considered an SG
 - In some cases an SG vs TS vs IS continuum needs to be considered
- Please weigh our opinion as part of your decision process
- direction@lists.isocpp.org.