

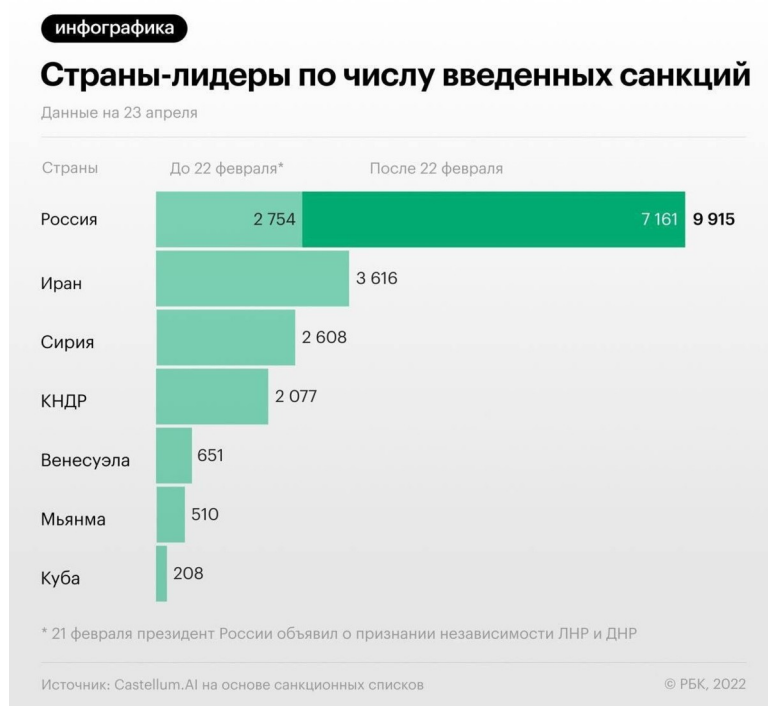
# Импортозамещай это

Соломенников Д.И., Техкомпания Хуавей, Санкт-Петербург

В статье рассматриваются вопросы импортозамещения важного технологического инструментария, используемого разработчиками программного обеспечения (языки программирования, компиляторы, операционные системы). Исследуются проблемы, связанные с лицензированием существующих инструментов. Предлагаются пути решения, в том числе, с помощью создания и поддержания сообществ разработчиков и разработки технологического инструментария на территории Российской Федерации.

## Введение

Наша страна испытывает небывалое санкционное давление. Многие видели графики, где сравнивается количество санкций, направленных на разные страны.



Есть что-то даже романтическое в том, что при 10000 ограничений мы продолжаем работать и многие даже не поняли, что вообще что-то изменилось. Честь и хвала тем людям, благодаря которым мы не сильно ощущаем последствия такого напора.

Тем не менее, вопрос с ограничениями довольно серьезный и легкомысленно к нему относиться не стоит.

В этой статье посмотрим на вопросы ограничений, касающихся языков программирования, компиляторов и вспомогательного инструментария, а также рассмотрим вопросы связанные с процессом разработки как таковым.

Для удобства дальнейшего повествования будем собирательно называть компиляторы, библиотеки и вспомогательный инструментарий «инструментами разработчика».

## Вводная информация

Небольшое введение для неспециалистов. Если вы знаете, что такое инструменты разработчика, этот раздел можно пропустить.

Итак, когда вы включаете компьютер, берете в руки планшет или телефон или запускаете станок с ЧПУ, все, что на нем работает — это программное обеспечение (ПО) или просто программы. Они могут быть очень разными по виду, выполняемым функциям, размеру и т.д. С точки зрения информационной индустрии, того самого ИТ, о котором в последнее время много говорят, это, по сути, конечный продукт. То есть, ПО разрабатывается, устанавливается, обслуживается, продается. Это товар, часто очень недешевый товар.

Для производства физических товаров используется различное оборудование, различные станки и технологические устройства. Для производства ПО также используются инструменты, которые с некоторой натяжкой можно назвать станками. Это и есть инструменты разработчика, «станки» в мире программирования.

Когда вы используете ПО, вы можете и не подозревать, что «станки» используются, но они есть всегда и их стоимость тем или иным образом включена в стоимость ПО.

## Изготовление инструментов разработчика

Инструменты разработчика, как и станки в физическом мире, как правило, сложны в изготовлении. Для их создания требуется высокая квалификация и продолжительное время.

Так, разработка нового языка и компилятора к нему с нуля занимает 4-7 лет силами таких компаний, как Apple и Google. Развитие же этого инструментария происходит и того дольше — язык C# празднует 20-летие<sup>1</sup> в этом году, языку Java — 27 лет.

Интересной особенностью такого «станкостроения» является то, что разработка ведется с привлечением сообщества. То есть компания-разработчик публикует исходные тексты под какой-нибудь открытой лицензией и приглашает неопределенный круг лиц и компании к доработке инструментария. Такая модель очень популярна и продуктивна. Над инструментами работают сотни и тысячи людей по всему миру. В том числе программисты из России.

## «Открытые» лицензии

Поскольку лицензия открытая, создается иллюзия вседоступности этого инструментария. Однако, это именно что иллюзия.

На деле же как использование инструментов разработчика, так и их разработка определяется содержимым лицензии. И во многих случаях там скрывается много такого, что заставляет задуматься.

Чтобы оставить контроль разработки за собой, компании, начавшие такую работу, могут просить сторонних разработчиков подписать документ о передаче права на доработку в пользу компании. То есть, разработка вроде бы и публичная, но все права на код остаются у компании-разработчика. Может использоваться схема, при которой в публичном поле есть не вся разработка, а часть инструментария может быть под другой, уже более строгой или

---

<sup>1</sup> <https://dotnet.microsoft.com/en-us/20th-anniversary>

платной лицензией. Также в лицензии может быть прописана необходимость делать отчисления с оборота, количества пользователей или объема выручки.

То есть, фактически инструменты разработчика находятся напрямую или опосредованно под контролем компании-разработчика. Эта ситуация, в целом, не является чем-то исключительным, в конце концов речь идет о бизнес-процессах и необходимости получать прибыль. Однако, это позволяет компаниям, создающим инструменты разработчика поступать совершенно не по-деловому.

Например, компания Oracle уже ограничивала<sup>2</sup> доступ к Java, а компания Microsoft в соответствии с американскими санкциями запретила возможность отправить пожертвования разработчикам для российских граждан и запрещает<sup>3</sup> поставлять GitHub Enterprise Server в Россию.

Повторюсь, хотя флагманские инструменты вроде компилятора или среды времени выполнения могут быть вполне себе свободными и не подвергаться строгим ограничениям сами по себе, простор для возможных ограничений весьма широк.

Курьезный, в общем-то, случай<sup>4</sup> с крошечной библиотекой `left-pad` показал, что последствия простого действия могут быть серьезные. Что уж говорить про целенаправленные действия.

## Меры противодействия

«Все импортозаместить невозможно, да и не нужно этого делать, но добиваться технологического суверенитета по критическим позициям судового оборудования, по самым значимым производственным процессам и технологиям — нужно»

— Президент России В. В. Путин.

Сейчас многое делается в ИТ для исправления ситуации. Меры нужные и важные. Хочу лишь подсветить тот факт, что и в этой области есть очень чувствительные и уязвимые места, в которых необходимо укреплять свои позиции и добиваться технологического суверенитета.

На мой взгляд, в рамках работ по улучшению ситуации в ИТ отрасли необходимо больше внимания уделять инфраструктурным вопросам. Эти задачи нужно ставить перед университетами, а также вовлекать коммерческие компании в эту деятельность. Важно заниматься развитием компетенций в целом, на первом этапе портируя и модифицируя то, что позволяет лицензия. Далее обязательно возникнет потребность в изобретении новых языков.

Создание новых языков программирования и компиляторов, хотя процесс и трудоемкий, но по силам небольшой команде опытных разработчиков. Наш опыт показывает, что за 2 года можно создать язык уровня Java и компилятор к нему силами коллектива из 3-4 опытных разработчиков и 5-6 интернов.

---

2 <https://www.zdnet.com/article/oracle-blocking-java-installs-in-russia/>

3 <https://docs.github.com/en/site-policy/other-site-policies/github-and-trade-controls#github-enterprise-server>

4 <https://techrocks.ru/2020/11/26/how-a-developer-broke-the-internet/>

Противоречия в сроках с указанными ранее (4-7 лет) нет, поскольку речь идет о языке и компиляторе, среда исполнения (обязательный элемент любой программы) может сильно отличаться по объему в зависимости от задачи и её создание требует больших трудозатрат.

Дальнейшая интеграция инструментов в процесс разработки может быть точечной и весьма продуктивной. Не всегда проекты уровня .Net нужны сразу и для широкой номенклатуры оборудования. Вполне возможен постепенный органичный рост и взросление инструментария.

## **Инфраструктура операционных систем**

Вопросы импортозамещения можно отнести и к операционным системам. Хотя операционные системы и не являются инструментами разработчика в полной мере, их можно отнести к инфраструктурным объектам.

Об операционных системах в части замещения можно говорить много. Попробуем лишь обозначить тему.

Развитие отечественных клонов Linux хорошо уменьшает зависимость от компании Microsoft<sup>5</sup>, но, поскольку ядро Linux контролируется корпорациями, в том числе и упомянутой Microsoft, ситуация здесь ничуть не лучше, чем с инструментами разработчика. Вопросы контроля исходных текстов, лицензирования, вопросы безопасности и наличие закладок в этой области стоят не менее остро.

На мой взгляд, работы по замещению стоит вести по нескольким направлениям.

Большее внимание другим операционным системам позволит нивелировать такую зависимость. На рынке представлено много других свободных ОС: FreeBSD (и вообще BSD-семейство), Haiku OS, React OS, RavnOS. Отрадно, что в этом же ряду стоят KasperskyOS<sup>6</sup> и Phantom OS<sup>7</sup> разрабатываемые в России.

Кроме того, российскому бизнесу стоит (возможно, через государственное участие) активнее участвовать в таких организациях, как Linux Foundation<sup>8</sup> или FreeBSD Foundation<sup>9</sup> для усиления влияния на операционные системы.

## **«Российский GitHub»**

Есть еще один вопрос, который можно смело считать инфраструктурным — то, как ведется работа с исходными текстами. В профессиональном сообществе это получило условное название «российский GitHub».

Снова небольшое пояснение для неспециалистов.

Программа собирается из исходных текстов и других файлов (картинки, звук, модели объектов и т. д.), называемых ресурсами. Для работы с ними применяются «системы контроля версий», позволяющие удобно управлять самими как исходными текстами, так

---

5 <https://lunduke.substack.com/p/microsofts-growing-control-of-linux>

6 <https://os.kaspersky.ru/>

7 <http://phantomos.org/>

8 <https://www.linuxfoundation.org/>

9 <https://freebsd.foundation.org/>

и ресурсами. Систем контроля версий существует довольно много, они могут быть развернуты на стороне предприятия либо арендуются у стороннего партнера.

Однако, управление исходниками и ресурсами — это не только, а зачастую и не столько вопросы работы непосредственно с файлами. Это, скорее, вопросы взаимодействия разработчиков между собой.

Так, на сервере системы контроля версий можно создавать карточки задач с описанием проблемы, комментировать, предлагать решения, проводить ревизию кода. На таких серверах есть механизмы разделения прав, позволяющие, например, только комментировать, проводить ревизию кода или разрешающие принимать решение и «вливать» его в исходный код.

Там же, рядом с исходным кодом и ресурсами могут размещаться базы знаний, связанных с разрабатываемым программным кодом, веб-страницы сопутствующих сайтов и так далее.

Это целый мир, сообщество разработчиков, где происходит общение и разработка.

Ценность такого общения зачастую бывает выше ценности самих программ.

Самый известный, хотя и далеко не единственный, сервер подобного рода — GitHub<sup>10</sup>, ныне принадлежащий компании Microsoft. И хотя компания на словах постулирует поддержку разработчиков, на деле подчиняется американским законам и без объяснения причин блокирует<sup>11</sup> разработчиков из России.

Решение этой проблемы — создание собственных аналогов подобных систем. Запрос на это среди российских разработчиков есть, и, судя по общению с коллегами, весьма чувствительный.

Работы в этом направлении вроде бы начаты, есть понимание<sup>12</sup>, что это делать нужно, но уж больно как-то неспешно.

При этом Китай с 2013 года разрабатывает и использует собственный<sup>13</sup> аналог сайта GitHub — Gitee<sup>14</sup>.

Россия находится сейчас даже в более сложном положении, чем Китай, и вопрос как хранения исходных кодов на территории страны, так и вопрос построения внутреннего сообщества разработчиков стоит очень остро. Это вопрос технологического суверенитета. Его обязательно нужно ставить и решать.

## Заключение

Инфраструктурные проекты — это не только мосты, дороги и аэропорты. Это еще и неприметные, но очень важные инструменты, используемые в повседневной работе большим количеством программистов.

Важно, чтобы о них тоже не забывали в процессе импортозамещения и развития технологического ландшафта нашей страны. Для этого требуется, в общем-то, немного.

---

10 <https://github.com/>

11 <https://vc.ru/dev/403440-github-zablokiroval-akkaunty-sbera-alfa-banka-i-nekotoryh-polzovateley-iz-rossii>

12 <https://habr.com/ru/news/t/692950/>

13 <https://techcrunch.com/2020/08/21/china-is-building-its-github-alternative-gitee/>

14 <https://gitee.com/>

Нужно обозначить потребность в развитии инструментов разработчика, операционных систем и сообщества на территории нашей страны с вовлечением академической среды и бизнеса в этот процесс.

## **Ключевые слова**

Импортозамещение, компилятор, язык программирования, операционная система, сообщество разработчиков.

# ***Solomennikov Dmitry: Replace it***

## **Keywords**

Software replacement, compiler, programming language, operation system, software developer community

## **Abstract**

The article discusses software replacement due to international sanctions. Author describes questions of software tools (programming languages, compiles, operation systems) development. Licensing problems of existing tools are reviewed. Possible solutions are proposed, including community development and support as well as tool development in Russia.