

Комментарий относительно **SPE-пирамиды** [программная инженерия]:

Structure-**P**atterns-**E**vents (**SPE**) пирамида как значимый фреймворк в системном мышлении определяет последовательный путь **E** → **P** → **S** анализа и понимания систем с учётом разных аспектов (элементов, факторов) систем и их взаимосвязей.

- **EVENTS LEVEL (EL)** – уровень событий

На этом уровне мы наблюдаем конкретные инциденты или отдельные проблемы, т.е. события, возникающие в процессе разработки ПО. Эти события, как правило, являются сиюминутными и наблюдаемыми (всеми участниками процесса) – например, к числу таких событий можно отнести

- сообщения о выявленных ошибках,
- проблемы с производительностью того или иного приложения, и т.п.

Такие события крайне важны и требуют к себе внимания, но быстрая ...

реакция на их появление, реализация действий в попытке их устранения *“сразу и сейчас”* не позволяет понять глубину той или иной возникшей проблемы, её **корневые причины** (англ. **root causes**).

- **PATTERNS LEVEL (PL)** – **уровень паттернов** [закономерностей]

Этот уровень подразумевает выявление тенденций и повторяющихся поведений, возникающих в результате анализа нескольких наблюдаемых событий с течением времени (**обратите внимание на фактор ВРЕМЕНИ!**). С точки зрения программной инженерии, примерами таких тенденций могут служить

- появления повторяющихся ошибок в последовательных релизах ПО, что может свидетельствовать о существующих проблемах в проектировании и реализации,
- наблюдаемое снижение производительности ПО на нескольких последовательных итерациях разработки, что может свидетельствовать о некоторых системных изъянах, и т.п.

Эти закономерности, если сравнивать их с тем, что происходит на уровне событий (**EL**), позволяет командам предвидеть, в той или иной степени, будущие проблемы и предпринимать определенные шаги (*а это уже комплекс мер*), а не просто реагировать на отдельные события спонтанно (следуя правилу *“увидел – сразу что-то сделал”*).

- **STRUCTURE LEVEL (SL)** – **уровень структуры** [системы, системных структур]

На этом уровне акцент делается на тех процессах, стратегиях и методологиях, которые могут влиять как на *события*, так и на *шаблоны*. Применительно к процессу разработки мышление на этом уровне может, например, включать в себя

- анализ применяемых методологий на предмет коммуникации в команде разработчиков, взаимодействия между командой разработчиков и заказчиками, подходы к улучшению продукта и пр.,
- рассмотрение используемых архитектурных решений (например, микросервисы против монолита), подходов к развертыванию, изме-

нений в инфраструктуре, языках разработки и др.,

- анализ особенностей координации работы команд, подходов к подготовке обновлений, реализации бизнес-процессов, принятию решений, и т.п.

Анализ подобных структурных аспектов позволяет выявить уже упомянутые выше **коренные причины** имеющихся проблем, а не только их наблюдаемые симптомы.

На этом же уровне делается попытка осмыслить те основные драйверы и существующие **ментальные модели** (**ММ** | сложившиеся модели восприятия действительности, которые формируют организационные аспекты работы в команде/компании), которые могут серьезно влиять, например, на

- понимание *‘хорошего качества’* продукта, необходимых усилий по его поддержанию, что может быть связано с практикой тестирования, по-

ниманием необходимости сочетать глубокое и автоматическое тестирование, оценкой рисков,

- культуру разработки (правила, стандарты, гайды, инструменты и практики), которая отдает предпочтение скорости разработки, а не (в меньшей степени) качеству кода, не уделяет достаточного внимания проверке исполнения процессов и их результатов, требований, касающихся выдерживанию сроков выполнения задач, аудиту,
- использование предыдущего опыта.

Критический анализ устоявшихся **ментальных моделей** способствует выработке 'обновленного' взгляда на возможные подходы к решению проблем и постепенному переходу к реализации изменений (улучшающих вмешательств).



ПРИМЕР 1 (SPE-pyramid) для рассмотрения в классе



- **EVENTS LEVEL (EL)** – уровень событий

Отслеживание ошибок

- ▶ Событие: Пользователь сообщает об ошибке, из-за которой происходит сбой приложения X при отправке формы.
- ▶ Действие (на уровне данного события): Команда разработчиков воспроизводит ошибку и делает исправление.

- **PATTERNS LEVEL (PL)** – уровень паттернов [закономерностей]

Повторяющиеся проблемы с производительностью приложения

- ▶ Шаблон: В нескольких релизах команда замечает, что проблемы с производительностью возникают во время периода пиковой нагрузки и после развертывания обновления функций.
- ▶ Действие: Команда проводит тестирование и анализ производительности, чтобы определить ведущие к данным проблемам общие факто-

ры, такие как повышенная нагрузка на базу данных или использование неэффективных алгоритмов (их реализаций).

- **STRUCTURE LEVEL (SL)** – **уровень структуры** [системы, системных структур]

Оценка процесса разработки

- ▶ Структура: Организация следует строгой линейной модели (модели водопада).
- ▶ Действие: Команда оценивает, может ли принятие методологии Agile улучшить реагирование на проблемы, выявленные на уровне событий, что позволит обеспечить непрерывную интеграцию, проверку стиля кода и регулярное тестирование на протяжении всего цикла разработки.

- + **ментальные модели (ММ)**

Имеющее предположение известной на рынке компании об обеспечимом качестве ПО (рост числа достаточно объемных и сложных про-

дуктов, разрабатываемых в условиях жестких требований, с большим числом взаимосвязанных компонентов и предназначенных для последующей работы при высокой нагрузке)

► Модель: В компании считают, что глубокое тестирование продуктов может быть отложено до конца цикла разработки, даже в условиях нынешних 'жестких' требований к ПО (существующая ММ).

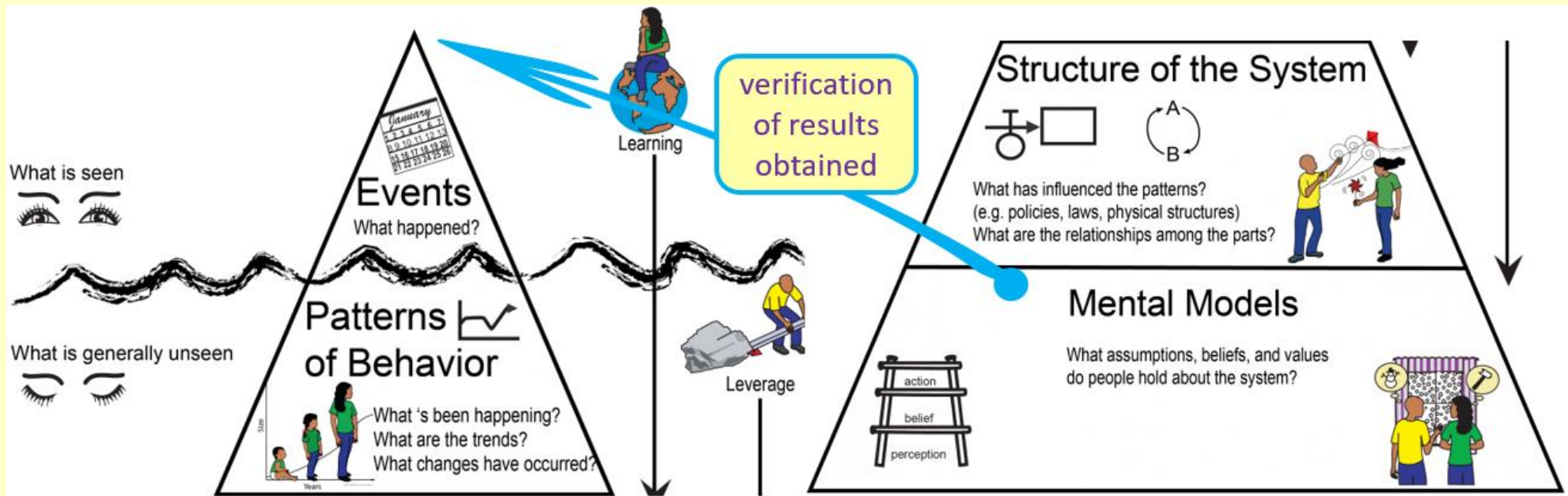
► Действие: Детально анализируя это предположение, принимая во внимание активное развитие CI/CD среды, компания постепенно переходит к пониманию необходимости применения в процессе разработки различных методов тестирования с привлечением и других специалистов, что должно привести в итоге к более быстрому выявлению ошибок и повышению качества ПО (наблюдается постепенный переход от [ментальной модели]₁ к [ментальной модели]₂).

Прикладной системный анализ – 1 курс [модуль 2, 04.12.2024]

	Action Mode	Time Orientation	Way of Perceiving	Questions You Would Ask
Events	React!	Present	Witness event	"What's the fastest way to react to this event now?"
Patterns	Adapt!		Measure or track patterns of events	"What kinds of trends or patterns of events seem to be recurring?"
Structure	Create Change!	Future	Causal loop diagrams and other systems thinking tools	"What structures are in place that are causing these patterns?"


Anderson V., Johnson L. Systems Thinking Basics, Pegasus Comm., 1997

Прикладной системный анализ – 1 курс [модуль 2, 04.12.2024]



Source:

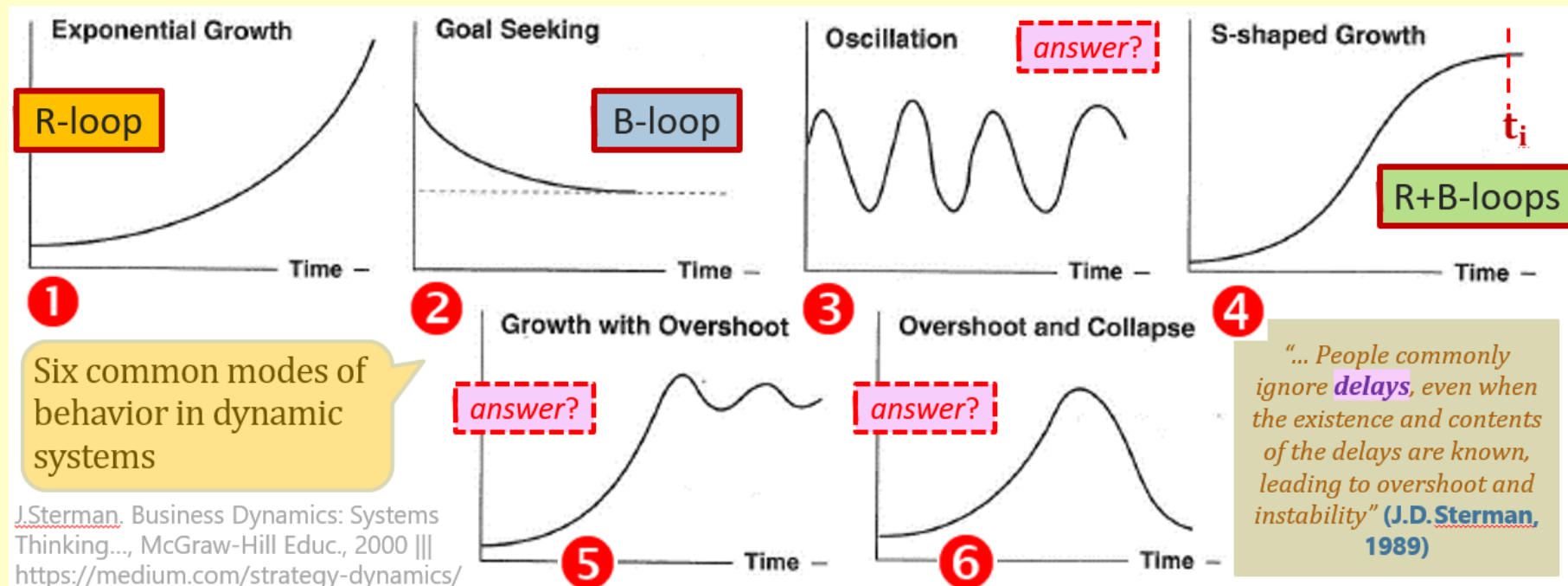
<https://www.watersfoundation.org/resources/iceberg-graphics/>

CLD allow to represent in the graphical form the *structure of the system* under consideration (they provide "... a language for articulating our understanding of dynamic, interconnected situations"); we say that *variables A* and *B* are causally related ()

CLD lay grounds for drawing various *hypotheses* connected to potential solutions (improvements) of the problem being represented as a system

CLD can be treated as *risk-free testers*

CLD are convenient tools to *document* (in clear and comprehensible form) the problem at hand (“... *they visualize variables and their relationships over time*” – recall BOT graph) ←



Прикладной системный анализ – 1 курс [модуль 2, 04.12.2024]

