## Anti-pattern

起源于design pattern的贬义术语，指部分专业过程中普遍存在的错误惯例。所有的anti-pattern都不仅仅是简单的错误，而是或多或少基于正确方法理论的错误/过度实践。

Analysis Paralysis —— 过分分析和思考，不实践

Cargo Cult Programming —— 不充分理解使用准则，盲目套用模块和代码

Death March —— 项目管理中，强行要求人员进行意识中注定失败或延期的任务

Groupthink —— 团队决策过程中，因为惧怕意见分歧而导致的非理性和谐

Vendor Lock-in —— 增加消费者切换服务商的代价以达到垄断目的

## Array of Structures(AoS)

最常见和传统的数据布局，记录record（也称struct结构体）只含一个对象的属性field。用结构体的数组来表示全部对象。适于遍历对象，但是不适于批量处理同一属性。

Structure of Arrays (SoA)

对比AoS，一个struct中含有全部对象的属性值，以平行数组方式存储。有别于常理，但是充分利用了数据局部性，不过不适于遍历对象。

Array of Structures of Arrays (AoSoA)

AoS和SoA的混合，将所有对象分组，大组即为结构体数组，每组结构体内以平行数组方式存储属性值。分组的尺寸一般由SIMD（single instruction multiple data）寄存器宽度决定。

## Data-oriented design

DOD或DOP是一种编程优化技巧，考虑到将相同类型、相同功能或相同处理的数据集中存储和处理，以提高程序的局部性并提高运算性能。

源数据和本地数据可以不同布局，源数据保持直观形式，本地数据则以便于计算的优化形式存储。

## Entity–component–system (ECS)

一种架构形式，主张component over inheritance，常用于游戏开发，如Unity。将对象视为实体entity，属性视为组件component。实体可灵活包含多种组件，实体之间不采用继承，好处是在运行时就可以修改组件以修改实体行为。经常和DOD一起使用，因为同类组件可以集中存储和处理。

ECS详细特征见wiki。

## Portable Network Graphics(PNG)

png格式一般是无损的（直接存储颜色值），但是有时对于相同颜色像素有压缩，且某些形式下有损。

jpeg是有损的（利用算法近似像素的颜色变化），且没有透明度alpha，适合于存放相片式photographic图片，png更适合存放文档或纯色样式的图片（即颜色变化陡峭）。

png文件由头header（8字节信息，不包含图片相关的关键信息）和接着的数个chunks组成，chunks有critical（图片本体关键内容，图片信息和图片本体等）和ancillary（附加可选内容，用以扩展功能）之分。

具体可见wiki，诸如如何压缩像素信息、chunks的详情。

## Resource acquisition is initialization(RAII)

这是C++为首的OOP语言所遵守的准则，对于习惯C++的人来说也是常识。当我们在函数中声明一个局部对象的时候，会自动调用构造函数进行对象的初始化，当整个main函数执行完成后，自动调用析构函数来销毁对象，整个过程无需人工介入，由操作系统自动完成。RAII不仅仅让程序员省去了必须记住销毁对象的麻烦，也被某些技术利用，如多线程的互斥锁实现。

## Shotgun debugging

盲目的非规程化的进行调试，如一次使用大量的可能方法，以期解决问题。