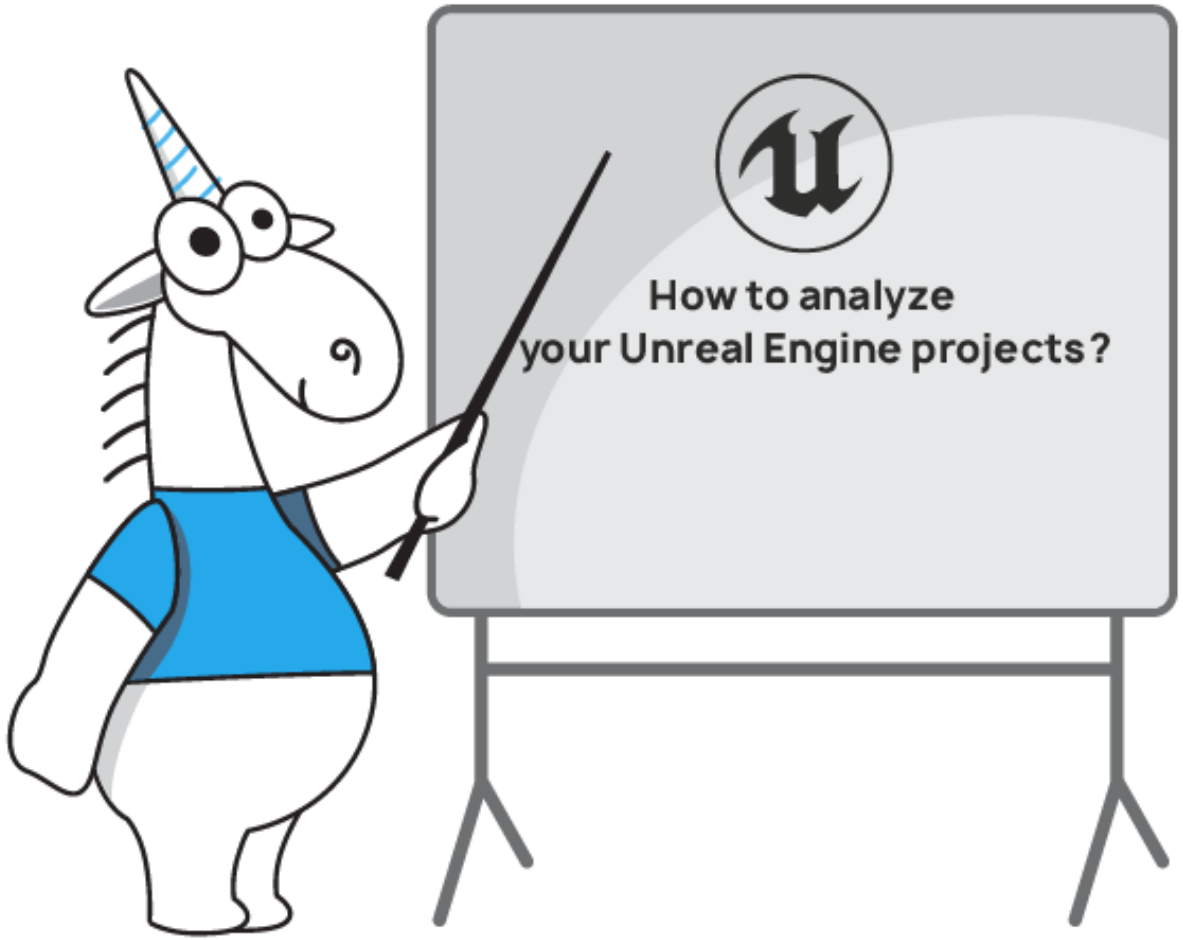
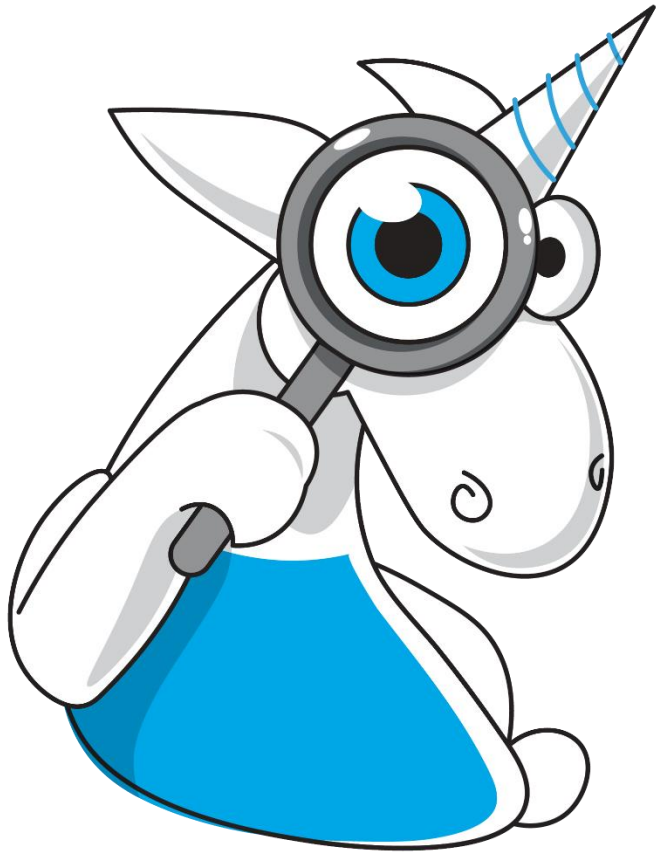
Статический анализ кода проектов, построенных на движке Unreal Engine

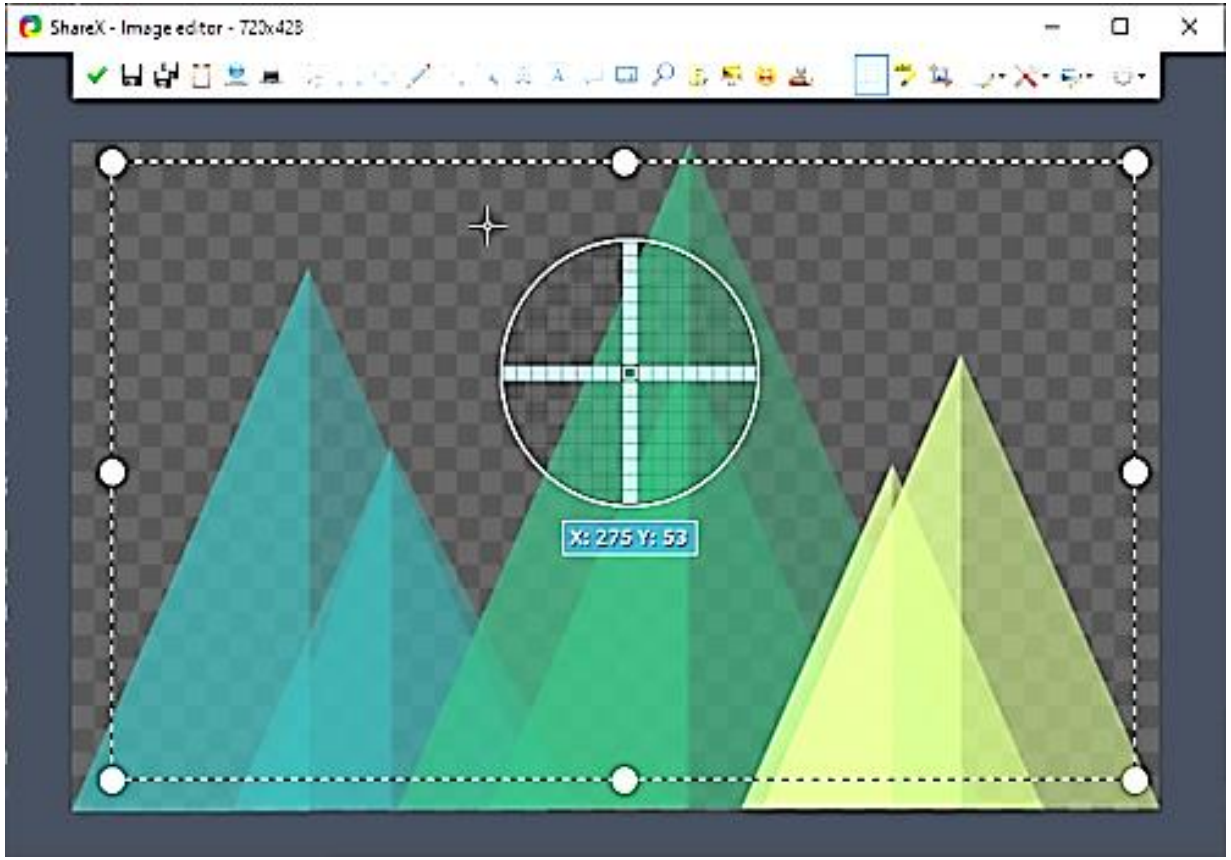


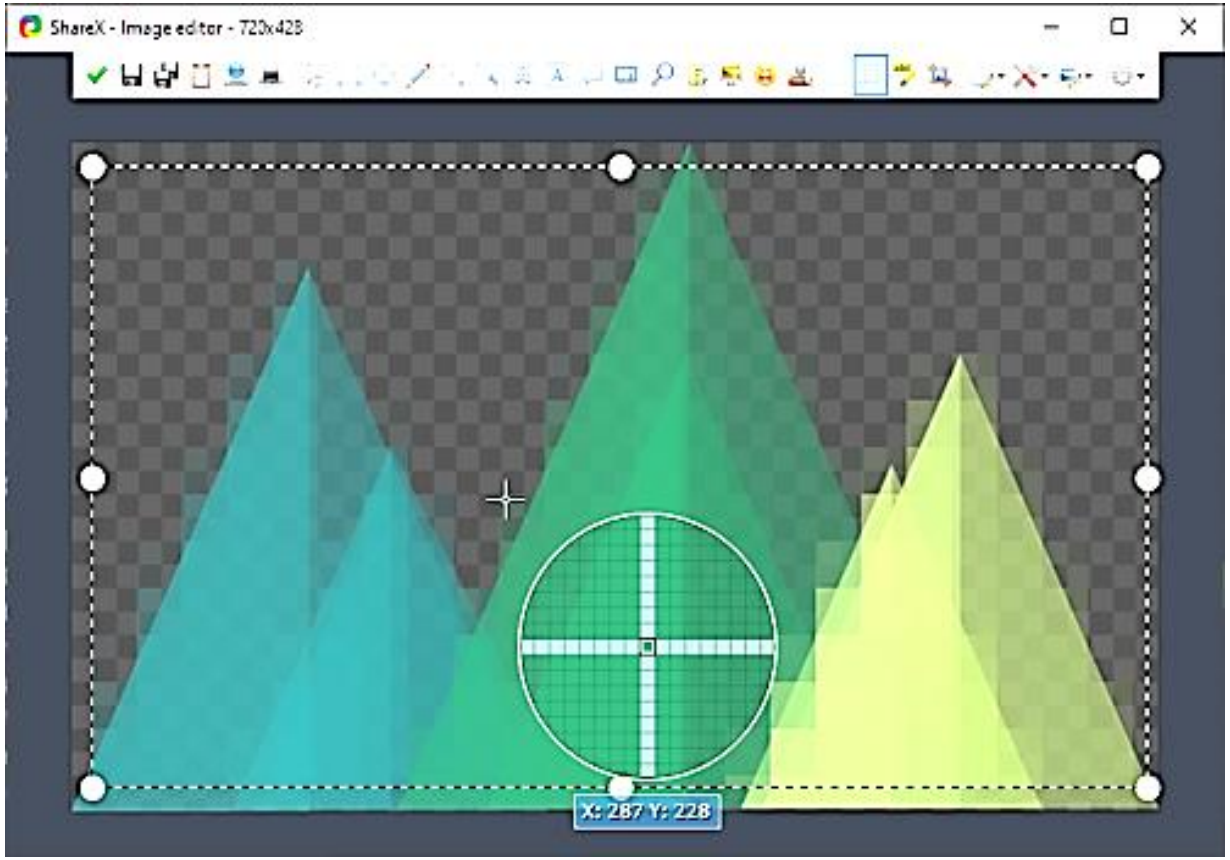
Илья Гайнулин
PVS-Studio
gainulin@viva64.com

Ищу ошибки в твоём коде



Результат неправильного приведения типов в проекте ShareX

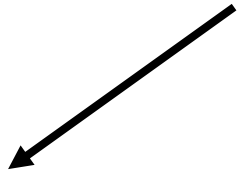




Результат неправильного приведения типов в проекте ShareX

тип `Byte`

Как было:

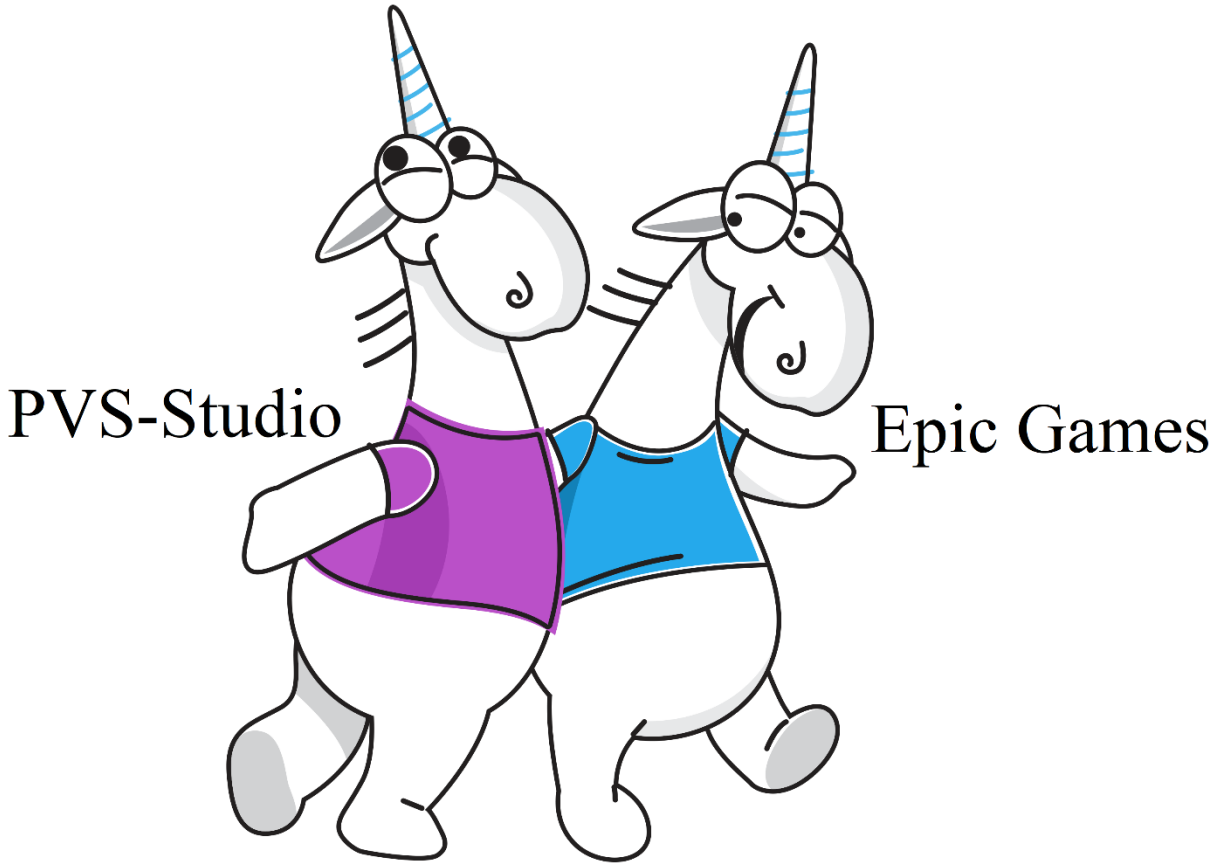


```
float pixelWeight = color.Alpha / 255;
```

Как поправили:

```
float pixelWeight = (float)color.Alpha / 255;
```

Сотрудничество с Epic Games



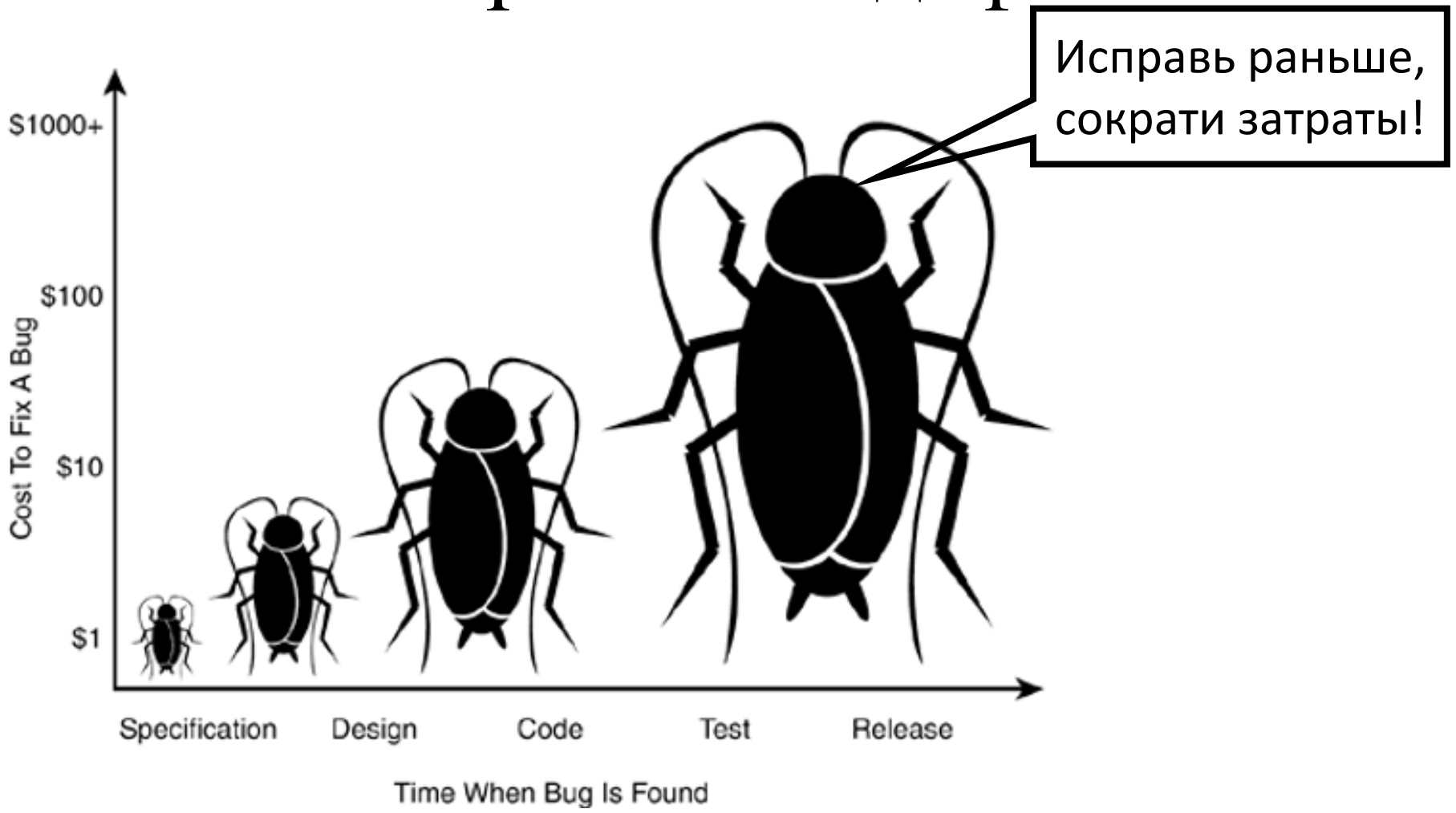
Что такое статический анализ?

Это анализ программного обеспечения, производимый без реального выполнения исследуемых программ.

Для чего нужен статический анализ?

- Обнаружение ошибок в коде на ранних этапах разработки программного обеспечения
- Получение рекомендаций по оформлению кода, проверка правописания
- Подсчёт различных метрик программного обеспечения

Стоимость исправления дефекта



Для чего нужен статический анализ?

- Обнаружение ошибок в коде на ранних этапах разработки программного обеспечения
- Получение рекомендаций по оформлению кода, проверка правописания
- Подсчёт различных метрик программного обеспечения

Преимущества статического анализа?

- Раннее обнаружение потенциальных проблем в коде
- Полное покрытие кода
- Простота использования

Преимущества статического анализа?

```
button_checked_gradient_begin = use_system_colors ? Color.Empty : Color.FromArgb (255, 223, 154);
button_checked_gradient_end = use_system_colors ? Color.Empty : Color.FromArgb (255, 166, 76);
button_checked_gradient_middle = use_system_colors ? Color.Empty : Color.FromArgb (255, 195, 116);
button_checked_highlight = Color.FromArgb (195, 211, 237);
button_checked_highlight_border = Color.FromKnownColor (KnownColor.Highlight);
button_pressed_border = use_system_colors ? Color.FromKnownColor (KnownColor.Highlight) : Color.FromArgb (0, 0, 128);
button_pressed_gradient_begin = use_system_colors ? Color.FromArgb (152, 181, 226) : Color.FromArgb (254, 128, 62);
button_pressed_gradient_end = use_system_colors ? Color.FromArgb (152, 181, 226) : Color.FromArgb (255, 223, 154);
button_pressed_gradient_middle = use_system_colors ? Color.FromArgb (152, 181, 226) : Color.FromArgb (255, 177, 109);
button_pressed_highlight = use_system_colors ? Color.FromArgb (150, 179, 225) : Color.FromArgb (150, 179, 225);
button_pressed_highlight_border = Color.FromKnownColor (KnownColor.Highlight);
button_selected_border = use_system_colors ? Color.FromKnownColor (KnownColor.Highlight) : Color.FromArgb (0, 0, 128);
button_selected_gradient_begin = use_system_colors ? Color.FromArgb (193, 210, 238) : Color.FromArgb (255, 255, 222);
button_selected_gradient_end = use_system_colors ? Color.FromArgb (193, 210, 238) : Color.FromArgb (255, 203, 136);
button_selected_gradient_middle = use_system_colors ? Color.FromArgb (193, 210, 238) : Color.FromArgb (255, 225, 172);
button_selected_highlight = use_system_colors ? Color.FromArgb (195, 211, 237) : Color.FromArgb (195, 211, 237);
button_selected_highlight_border = use_system_colors ? Color.FromKnownColor (KnownColor.Highlight) : Color.FromArgb (0, 0, 128);

check_background = use_system_colors ? Color.FromKnownColor (KnownColor.Highlight) : Color.FromArgb (255, 192, 111);
check_pressed_background = use_system_colors ? Color.FromArgb (152, 181, 226) : Color.FromArgb (254, 128, 62);
check_selected_background = use_system_colors ? Color.FromArgb (152, 181, 226) : Color.FromArgb (254, 128, 62);

grip_dark = use_system_colors ? Color.FromArgb (193, 190, 179) : Color.FromArgb (39, 65, 118);
grip_light = use_system_colors ? SystemColors.Window : Color.FromArgb (255, 255, 255);

image_margin_gradient_begin = use_system_colors ? Color.FromArgb (251, 250, 246) : Color.FromArgb (227, 239, 255);
image_margin_gradient_end = use_system_colors ? SystemColors.Control : Color.FromArgb (123, 164, 224);
image_margin_gradient_middle = use_system_colors ? Color.FromArgb (246, 244, 236) : Color.FromArgb (203, 225, 252);
image_margin_revealed_gradient_begin = use_system_colors ? Color.FromArgb (247, 246, 239) : Color.FromArgb (203, 221, 246);
image_margin_revealed_gradient_end = use_system_colors ? Color.FromArgb (238, 235, 220) : Color.FromArgb (114, 155, 215);
image_margin_revealed_gradient_middle = use_system_colors ? Color.FromArgb (242, 240, 228) : Color.FromArgb (161, 197, 249);
```

Преимущества статического анализа?

```
button_checked_gradient_begin = use_system_colors ? Color.Empty : Color.FromArgb (255, 223, 154);
button_checked_gradient_end = use_system_colors ? Color.Empty : Color.FromArgb (255, 166, 76);
button_checked_gradient_middle = use_system_colors ? Color.Empty : Color.FromArgb (255, 195, 116);
button_checked_highlight = Color.FromArgb (195, 211, 237);
button_checked_highlight_border = Color.FromKnownColor (KnownColor.Highlight);
button_pressed_border = use_system_colors ? Color.FromKnownColor (KnownColor.Highlight) : Color.FromArgb (0, 0, 128);
button_pressed_gradient_begin = use_system_colors ? Color.FromArgb (152, 181, 226) : Color.FromArgb (254, 128, 62);
button_pressed_gradient_end = use_system_colors ? Color.FromArgb (152, 181, 226) : Color.FromArgb (255, 223, 154);
button_pressed_gradient_middle = use_system_colors ? Color.FromArgb (152, 181, 226) : Color.FromArgb (255, 177, 109);
button_pressed_highlight = use_system_colors ? Color.FromArgb (150, 179, 225) : Color.FromArgb (150, 179, 225);
button_pressed_highlight_border = Color.FromKnownColor (KnownColor.Highlight);
button_selected_border = use_system_colors ? Color.FromKnownColor (KnownColor.Highlight) : Color.FromArgb (0, 0, 128);
button_selected_gradient_begin = use_system_colors ? Color.FromArgb (193, 210, 238) : Color.FromArgb (255, 255, 222);
button_selected_gradient_end = use_system_colors ? Color.FromArgb (193, 210, 238) : Color.FromArgb (255, 203, 136);
button_selected_gradient_middle = use_system_colors ? Color.FromArgb (193, 210, 238) : Color.FromArgb (255, 225, 172);
button_selected_highlight = use_system_colors ? Color.FromArgb (195, 211, 237) : Color.FromArgb (195, 211, 237);
button_selected_highlight_border = use_system_colors ? Color.FromKnownColor (KnownColor.Highlight) : Color.FromArgb (0, 0, 128);

check_background = use_system_colors ? Color.FromKnownColor (KnownColor.Highlight) : Color.FromArgb (255, 192, 111);
check_pressed_background = use_system_colors ? Color.FromArgb (152, 181, 226) : Color.FromArgb (254, 128, 62);
check_selected_background = use_system_colors ? Color.FromArgb (152, 181, 226) : Color.FromArgb (254, 128, 62);

grip_dark = use_system_colors ? Color.FromArgb (193, 190, 179) : Color.FromArgb (39, 65, 118);
grip_light = use_system_colors ? SystemColors.Window : Color.FromArgb (255, 255, 255);

image_margin_gradient_begin = use_system_colors ? Color.FromArgb (251, 250, 246) : Color.FromArgb (227, 239, 255);
image_margin_gradient_end = use_system_colors ? SystemColors.Control : Color.FromArgb (123, 164, 224);
image_margin_gradient_middle = use_system_colors ? Color.FromArgb (246, 244, 236) : Color.FromArgb (203, 225, 252);
image_margin_revealed_gradient_begin = use_system_colors ? Color.FromArgb (247, 246, 239) : Color.FromArgb (203, 221, 246);
image_margin_revealed_gradient_end = use_system_colors ? Color.FromArgb (238, 235, 220) : Color.FromArgb (114, 155, 215);
image_margin_revealed_gradient_middle = use_system_colors ? Color.FromArgb (242, 240, 228) : Color.FromArgb (161, 197, 249);
```

Преимущества статического анализа?

```
button_pressed_highlight = use_system_colors ?
                           Color.FromArgb (150, 179, 225) :
                           Color.FromArgb (150, 179, 225);
```

Mono

V3012 The '?:' operator, regardless of its conditional expression, always returns one and the same value: Color.FromArgb (150, 179, 225). ProfessionalColorTable.cs 258

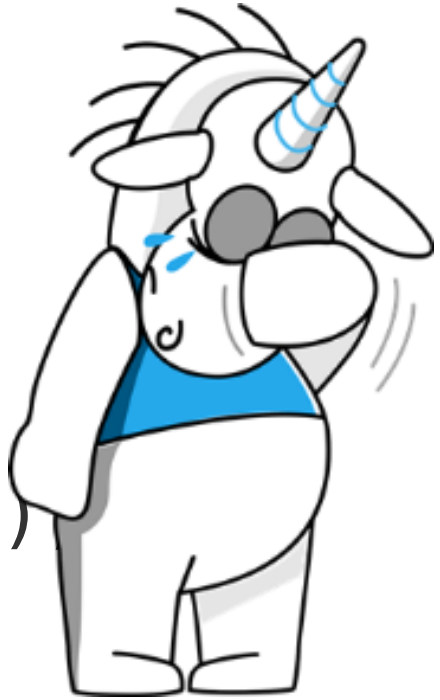
Недостатки статического анализа

- Неизбежное появление ложно-позитивных срабатываний
- Статический анализ, как правило, слаб в диагностике утечек памяти и параллельных ошибок

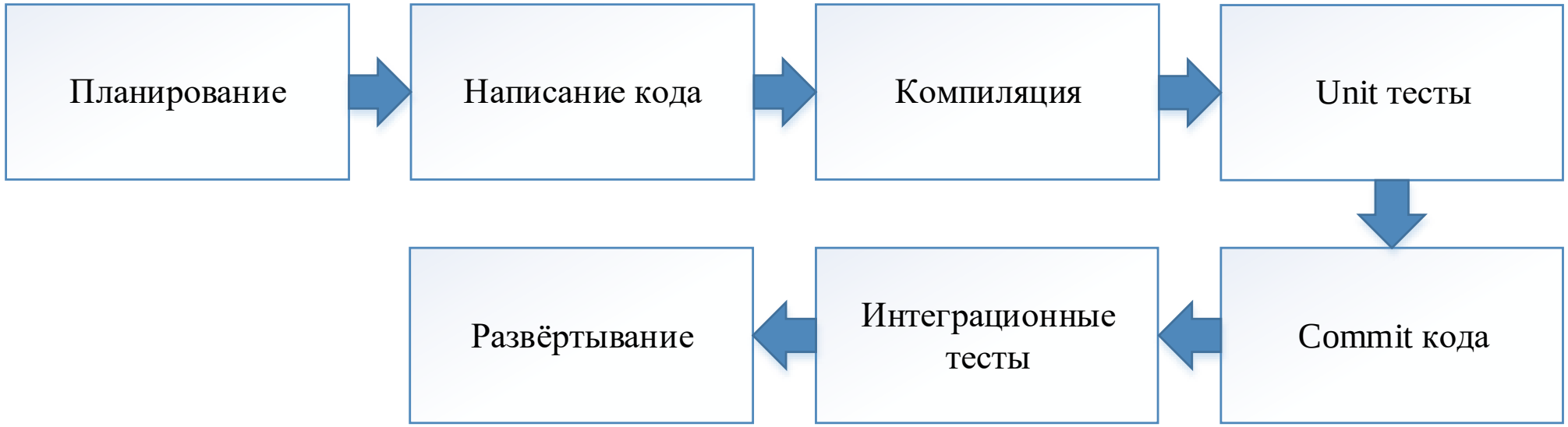
Недостатки статического анализа

```
void OutstandingIssue(const char *strCount)
{
  unsigned nCount;
  sscanf_s(strCount, "%u", &nCount);

  int array[10];
  memset(array, 0, nCount * sizeof(int));
}
```

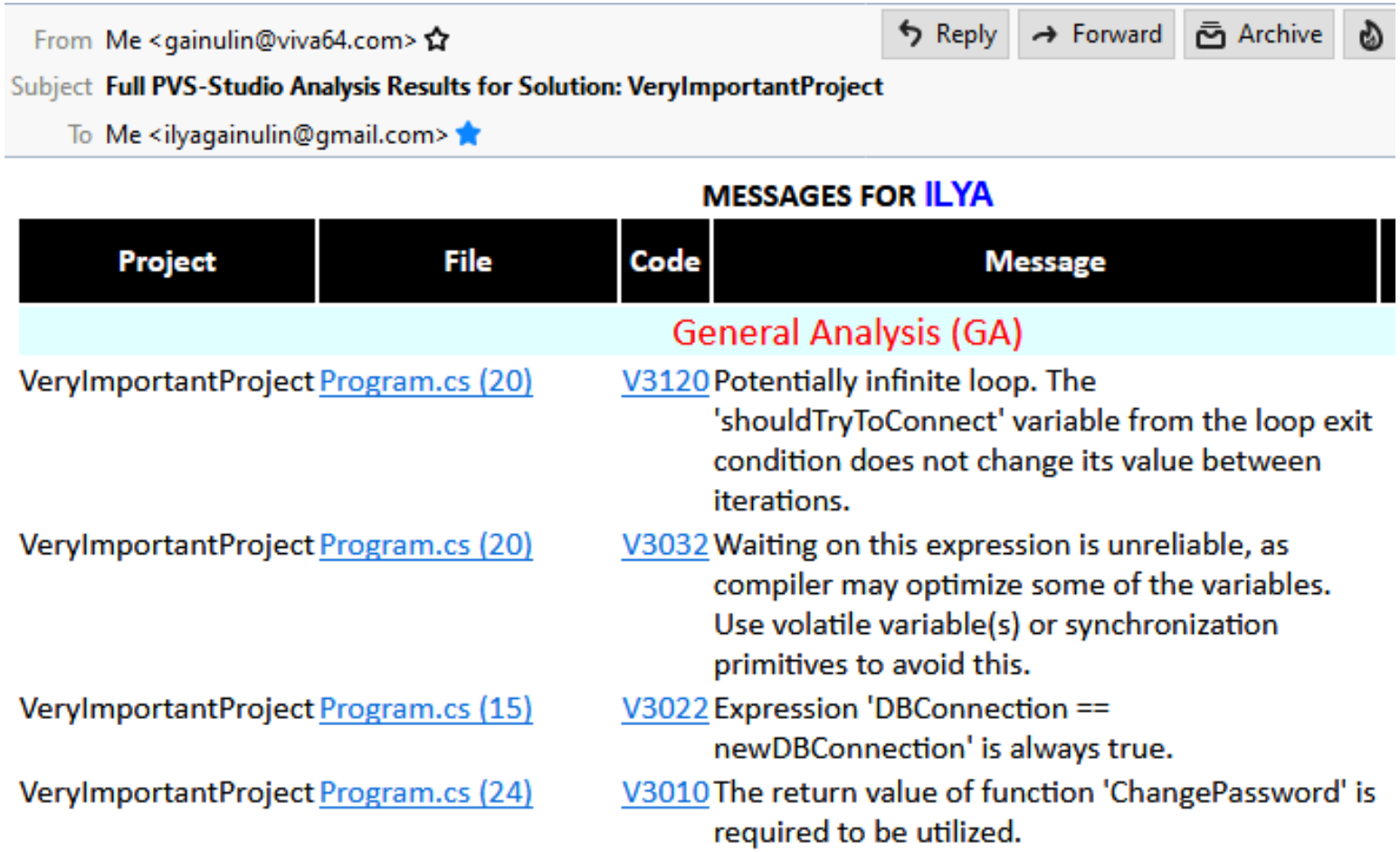


Статические анализаторы — это не изолированные инструменты и редкие проверки, а часть DevOps



Статические анализаторы — это не изолированные инструменты и редкие проверки, а часть DevOps

Статические анализаторы — это не изолированные инструменты и редкие проверки, а часть DevOps



From Me <gainulin@viva64.com>
Subject **Full PVS-Studio Analysis Results for Solution: VeryImportantProject**
To Me <ilyagainulin@gmail.com>

Reply Forward Archive

MESSAGES FOR ILYA

Project	File	Code	Message
			General Analysis (GA)
VeryImportantProject	Program.cs (20)	V3120	Potentially infinite loop. The 'shouldTryToConnect' variable from the loop exit condition does not change its value between iterations.
VeryImportantProject	Program.cs (20)	V3032	Waiting on this expression is unreliable, as compiler may optimize some of the variables. Use volatile variable(s) or synchronization primitives to avoid this.
VeryImportantProject	Program.cs (15)	V3022	Expression 'DBConnection == newDBConnection' is always true.
VeryImportantProject	Program.cs (24)	V3010	The return value of function 'ChangePassword' is required to be utilized.

Статические анализаторы — это не изолированные инструменты и редкие проверки, а часть DevOps

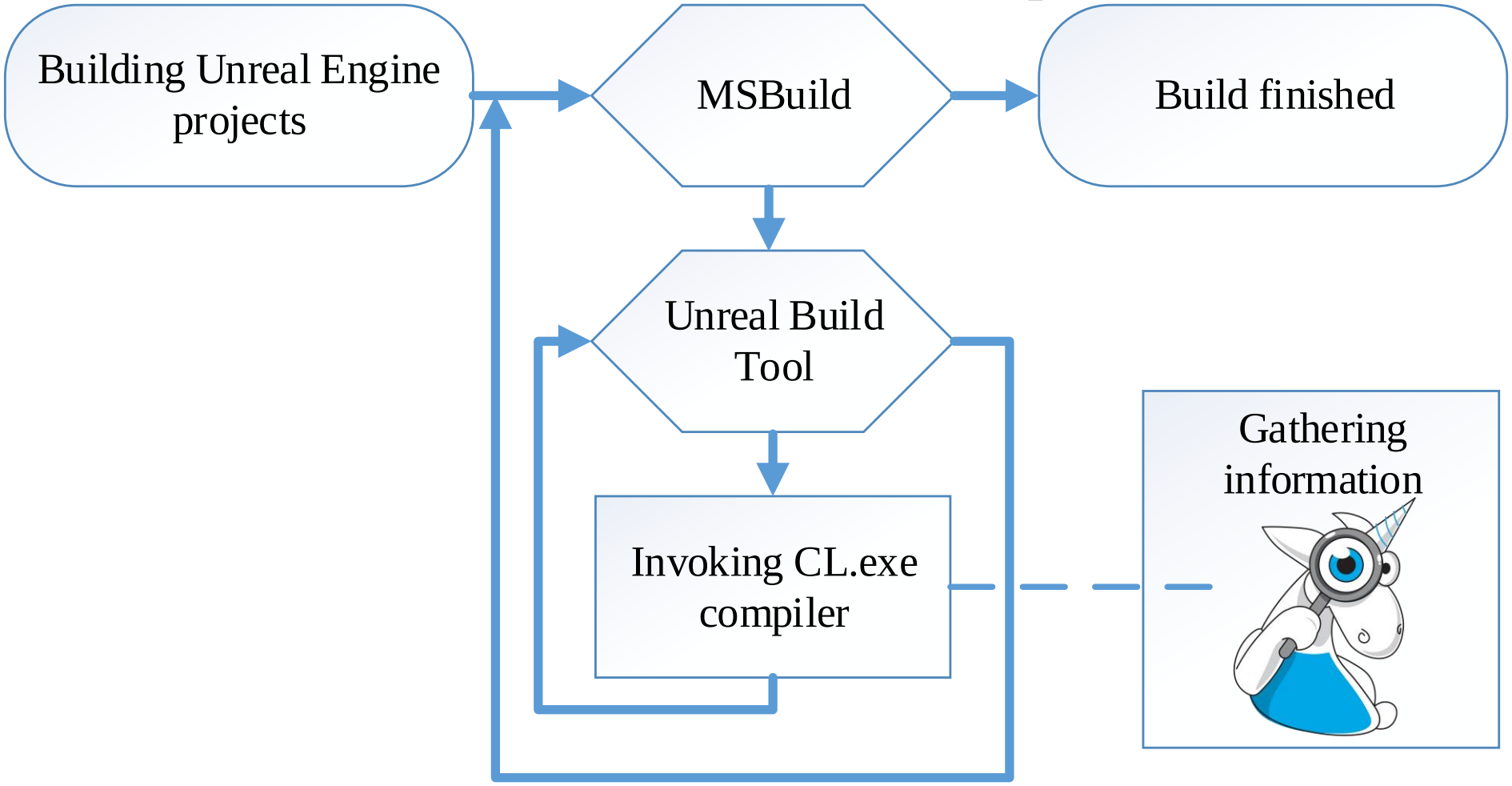
Инкрементальный анализ

- Сокращение времени анализа, так как анализируется только исправленный или новый код
- Хорошо подходит для раннего обнаружения ошибок

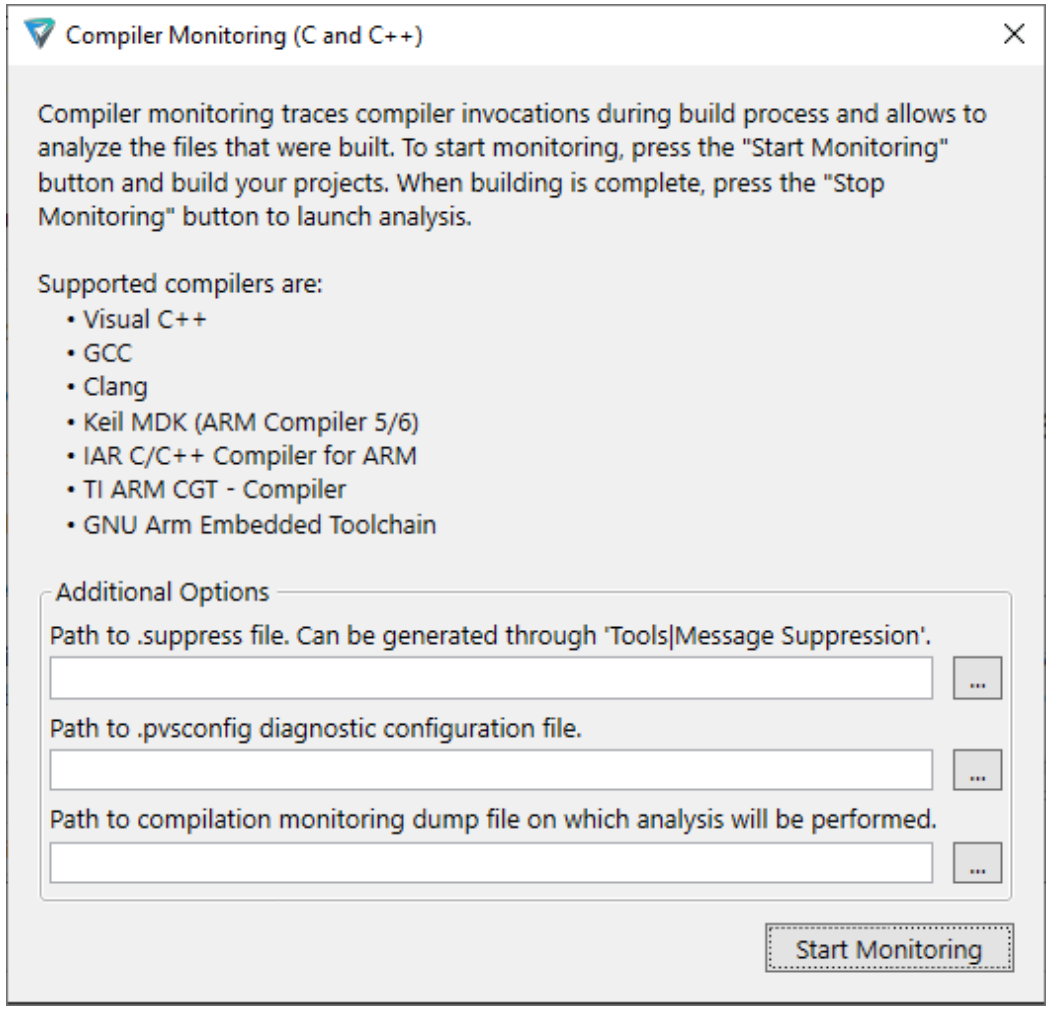
Способы проведения статического анализа UE проекта

- Отслеживания запусков компилятора
- Прямая интеграция с системой сборки

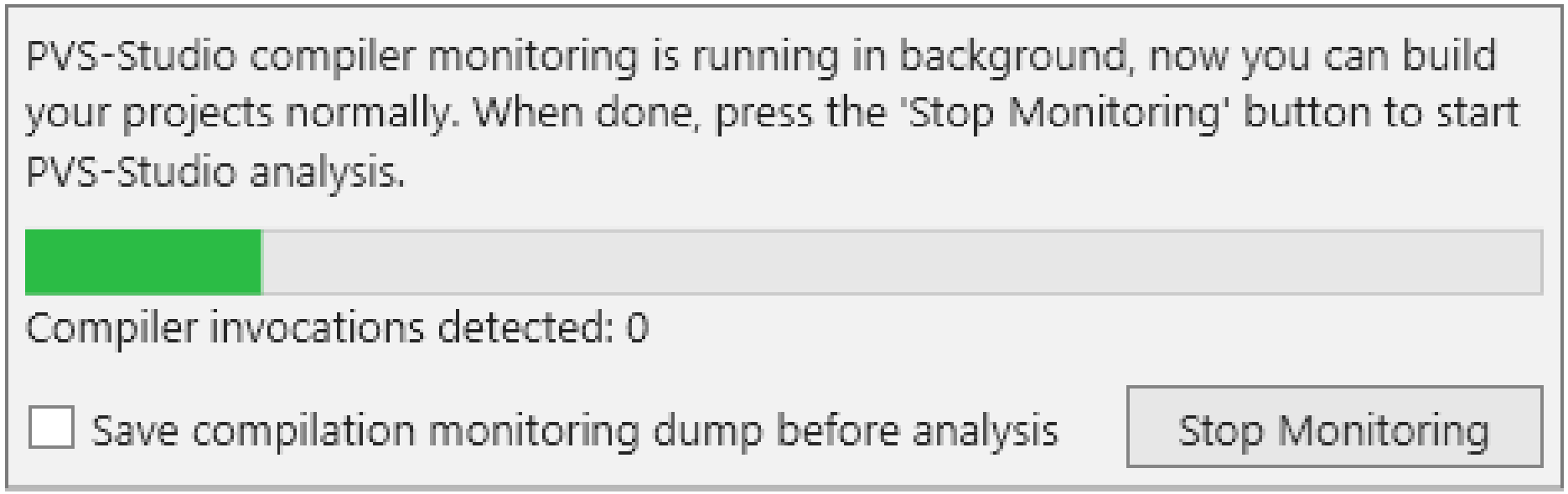
Самый простой способ проверить UE проект – это отловить все вызовы компилятора



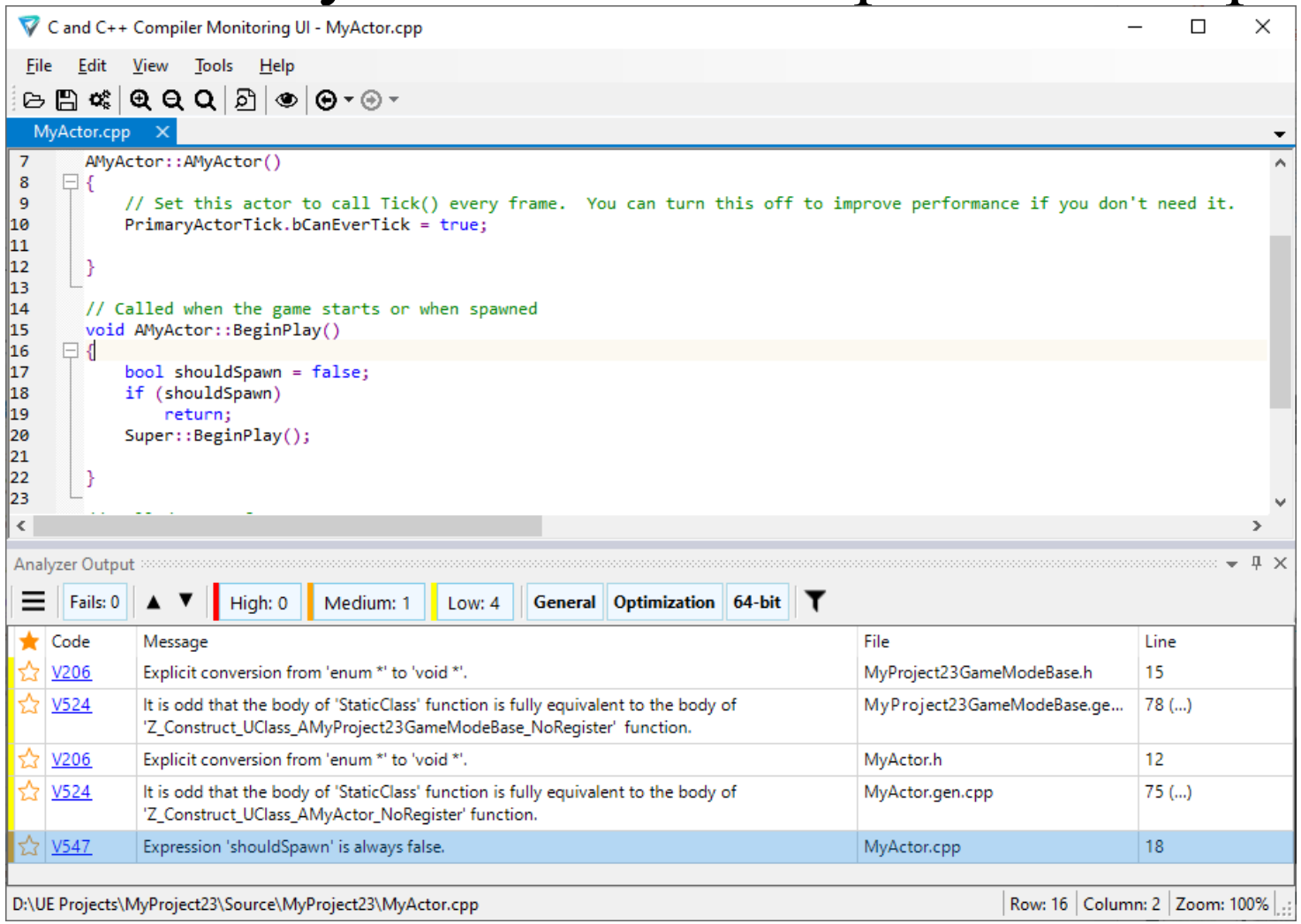
Отслеживание запусков компилятора из GUI приложения



Отслеживание запусков компилятора из GUI приложения

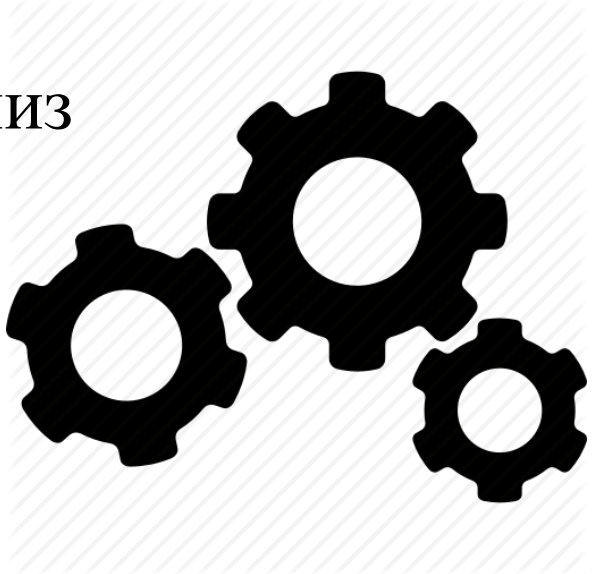


Отслеживание запусков компилятора из GUI приложения

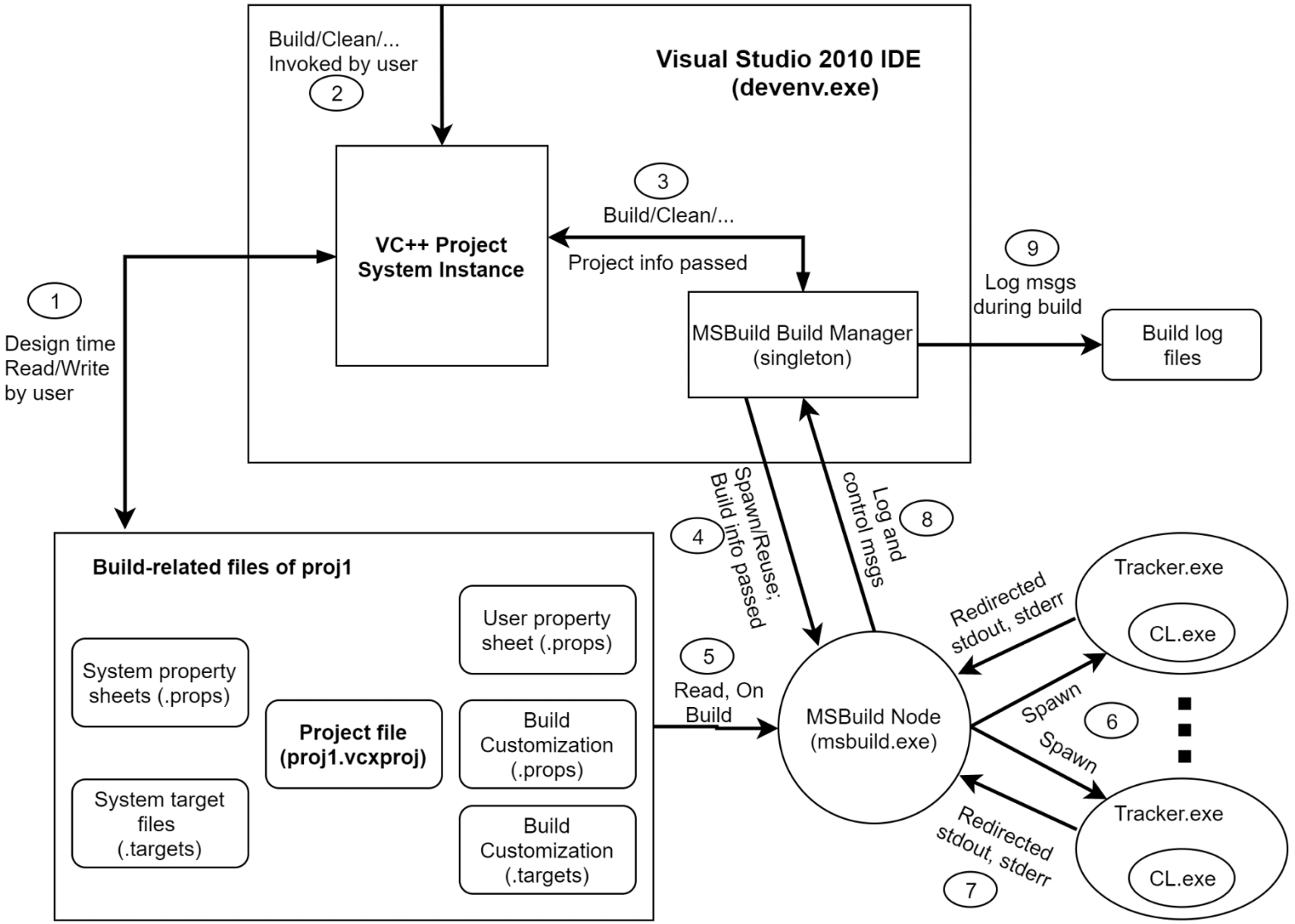


Прямая интеграция в сборочную систему и VS

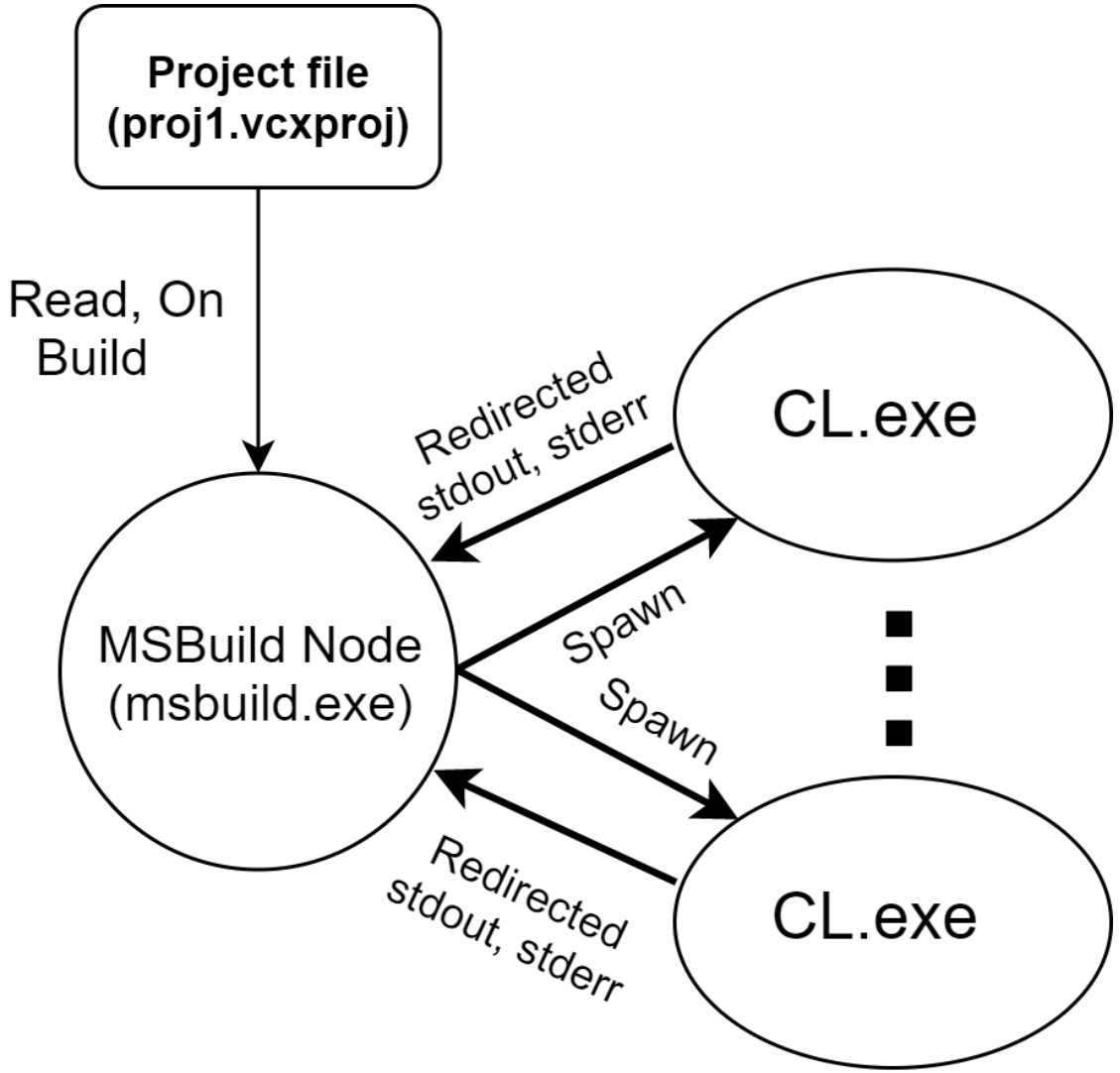
- Возможность проводить анализ напрямую из IDE Visual Studio
- Автоматический запуск анализа при осуществлении сборки UE проекта
- Возможность осуществлять инкрементальный анализ



Обычный сценарий сборки C++ проекта в VS

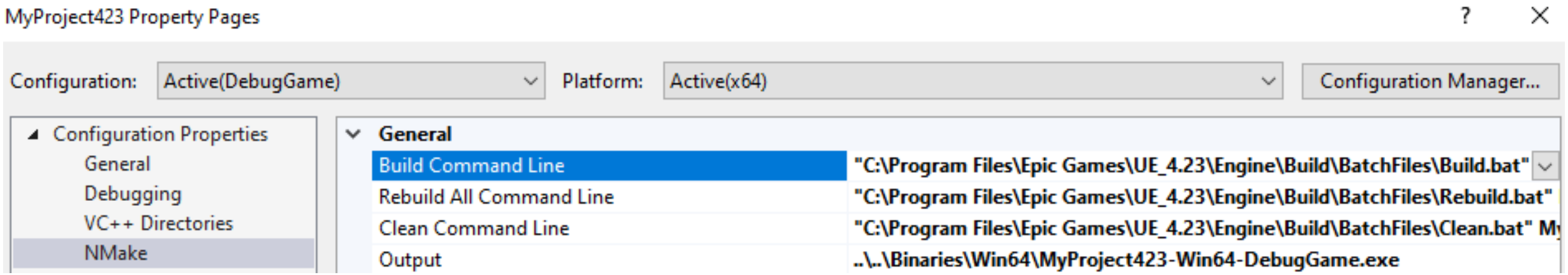


Обычный сценарий сборки C++ проекта в VS

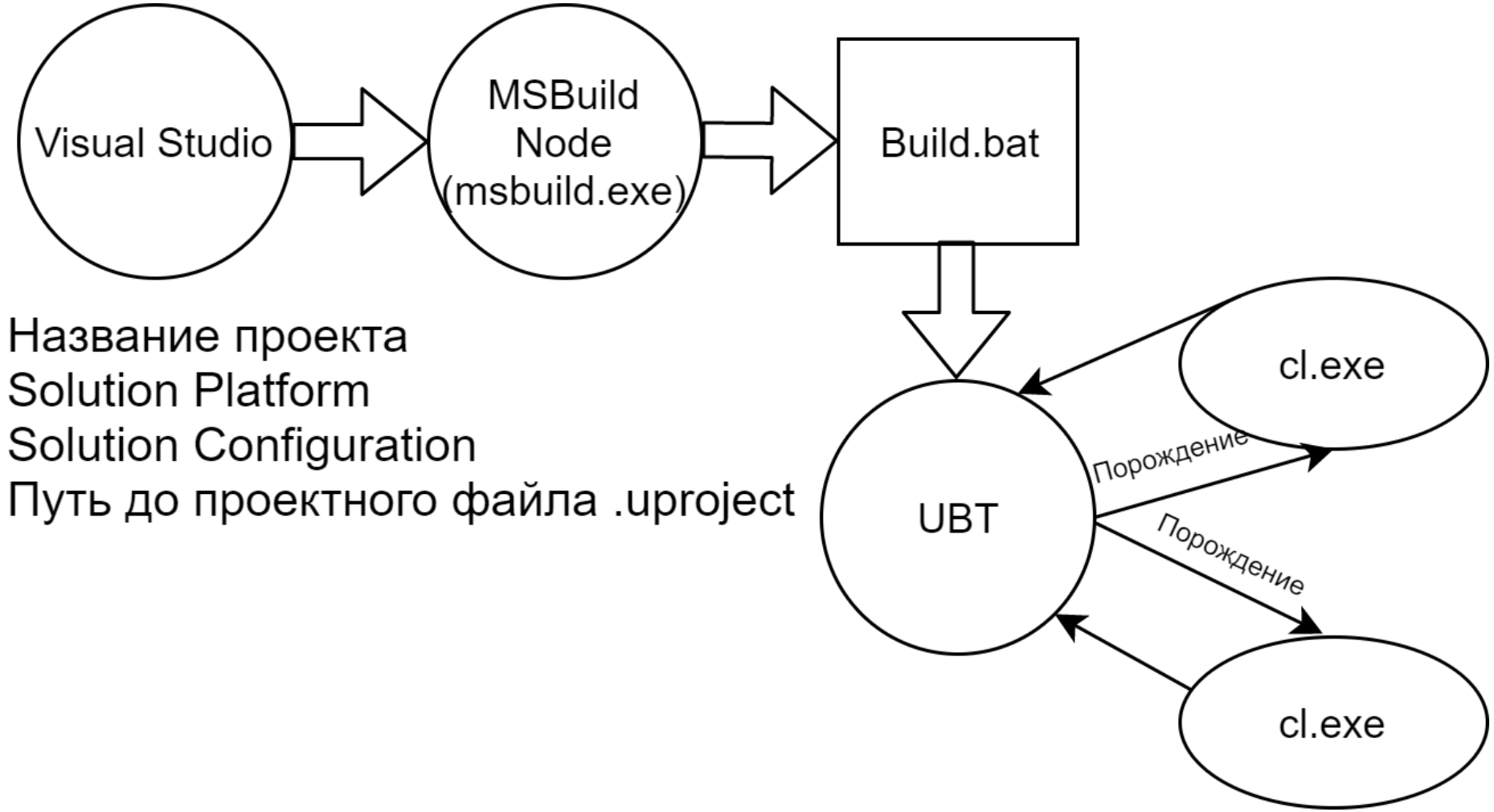


Особенность сборки UE проекта

- Использование не стандартной сборочной системы для Visual Studio - MSBuild, а собственной – UnrealBuildTool (UBT)
- Использование проектных файлов с расширением .uproject



Особенность сборки UE проекта



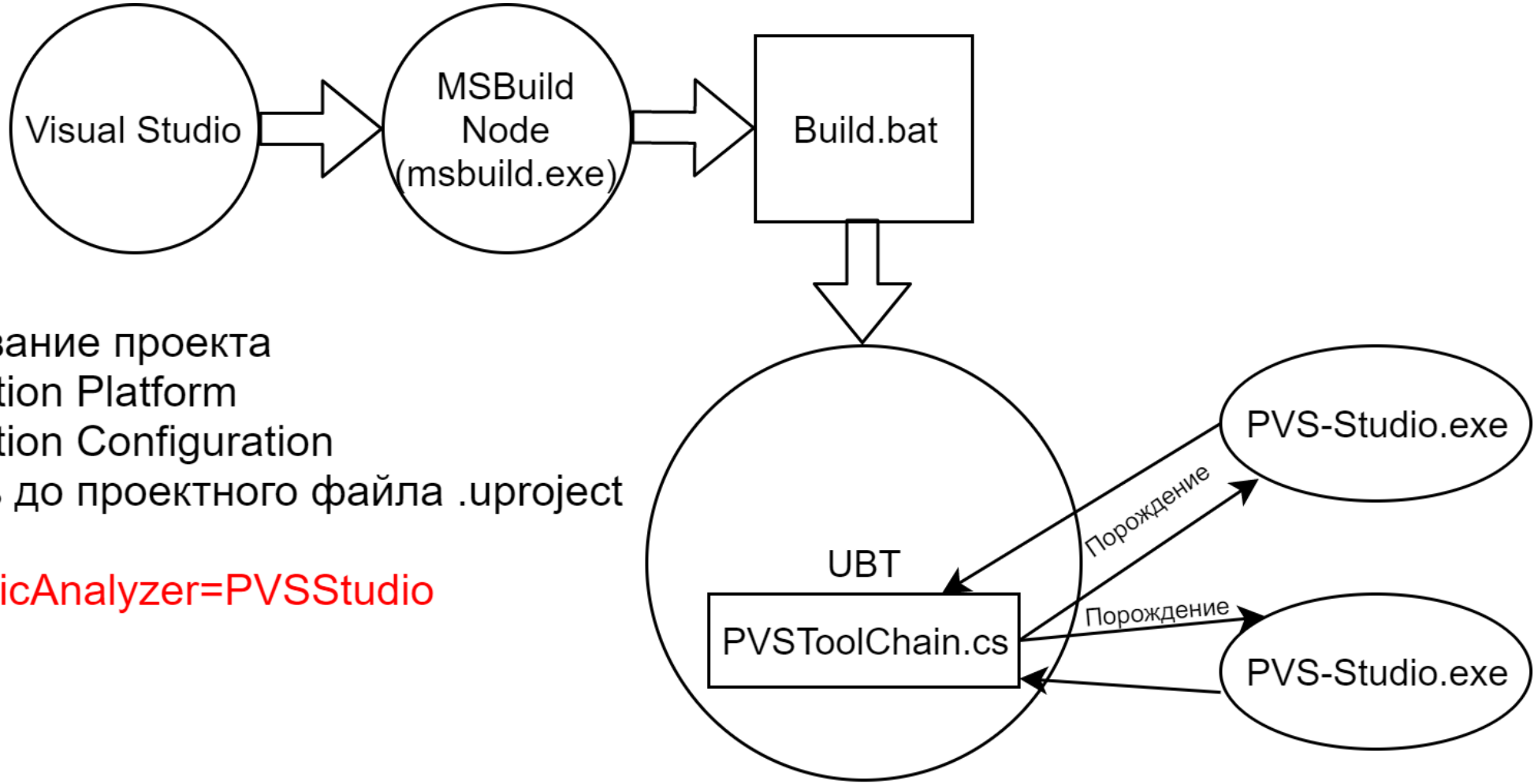
В чём проблема?

- В проектном файле могут отсутствовать пути до исходных файлов, которые необходимо проверять
- Присутствуют не все параметры компиляции, необходимые для правильного препроцессирования

Какое решение?

- Встроить в UBT запуск статического анализатора, то есть написать свой toolchain для сборочной системы
- Передавать какой-нибудь параметр сборочной системе UBT при запуске, указывающий о необходимости проведения анализа

Какое решение?



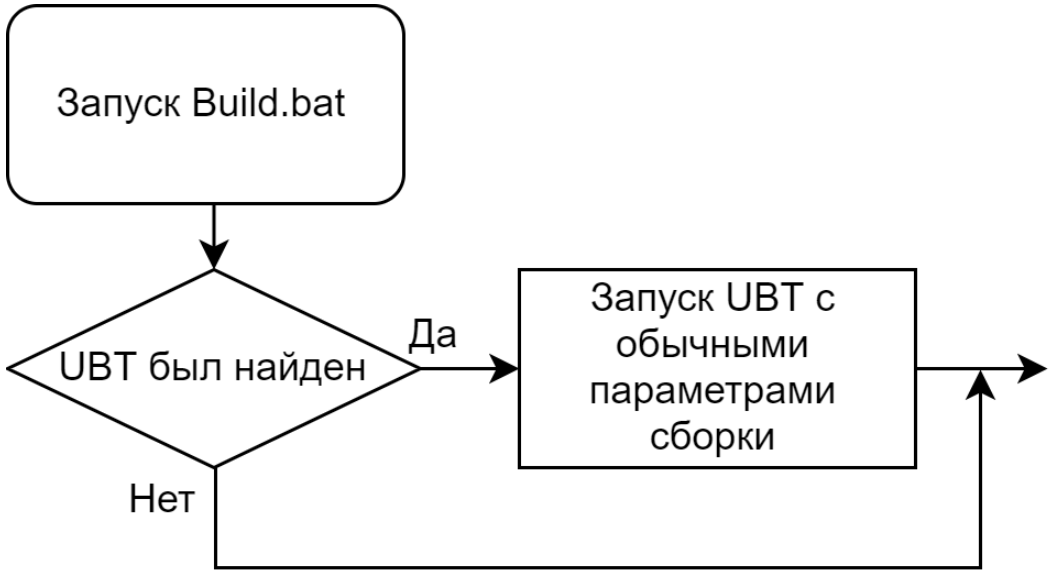
Несколько ограничений существующего toolchain’a

- Только сборка или только анализ проекта
- Нет анализа .cpp файла, если был изменён подключённый .h файл

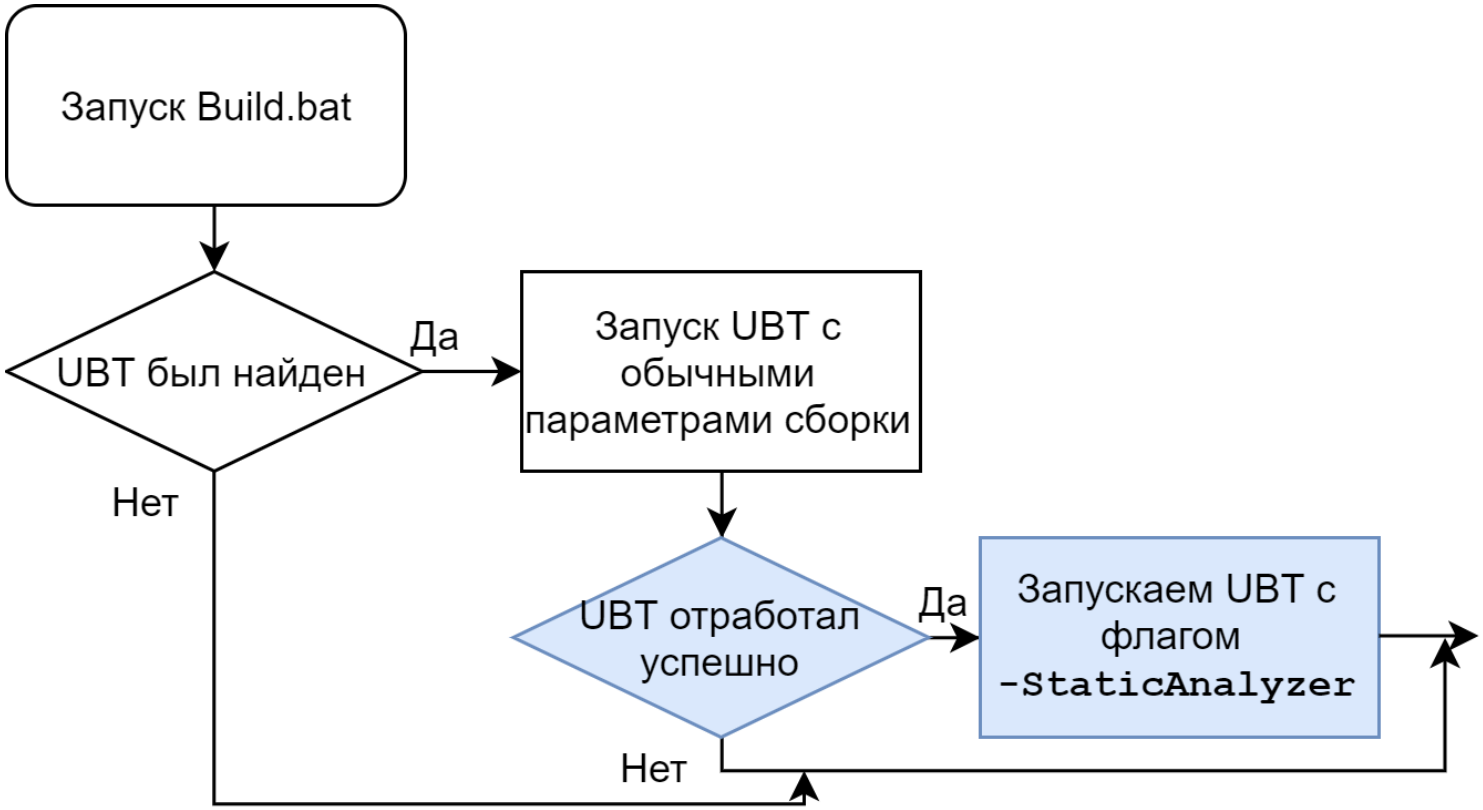
Анализ проекта после его сборки

- Модификация скриптов Build.bat и Rebuild.bat

Анализ проекта после его сборки



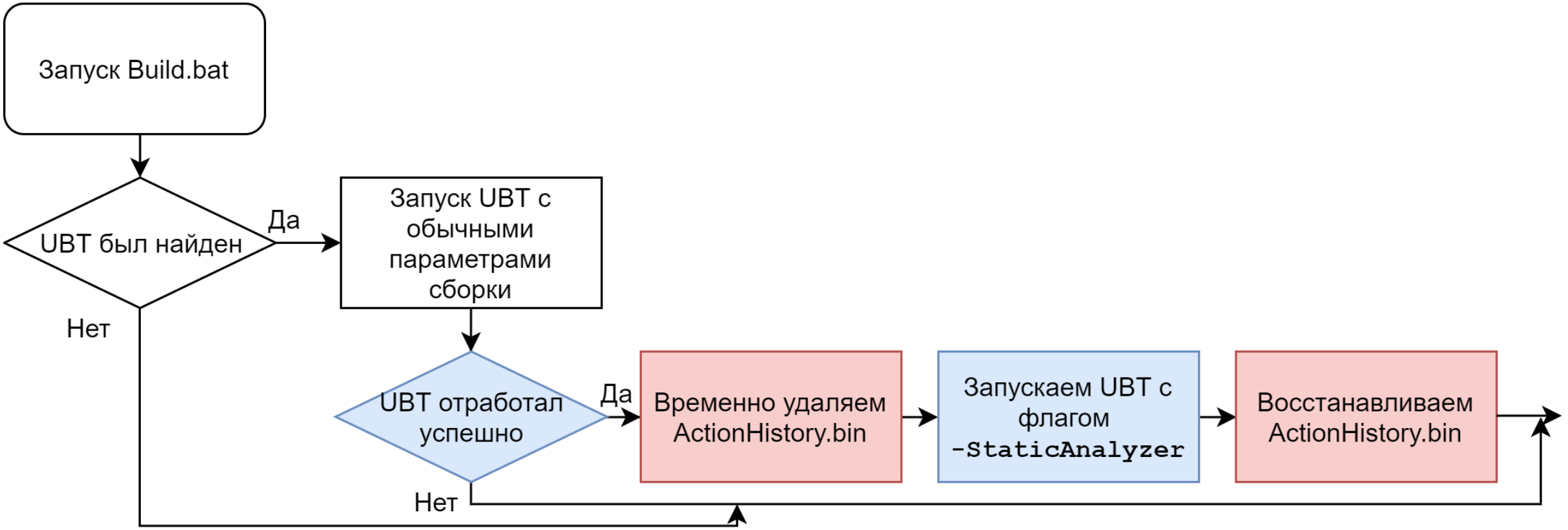
Анализ проекта после его сборки



Анализ .cpp файла при модификации подключённого .h

- Удаление и восстановление кэш файла, используемого UBT

Анализ .cpp файла при модификации подключённого .h



Примеры ошибок в коде движка Unreal Engine

```
static bool PositionIsInside(....)
{
  return
    Position.X >= Control.Center.X - BoxSize.X * 0.5f &&
    Position.X <= Control.Center.X + BoxSize.X * 0.5f &&
    Position.Y >= Control.Center.Y - BoxSize.Y * 0.5f &&
    Position.Y >= Control.Center.Y - BoxSize.Y * 0.5f;
}
```

Примеры ошибок в коде движка Unreal Engine

```
static bool PositionIsInside(....)
{
  return
    Position.X >= Control.Center.X - BoxSize.X * 0.5f &&
    Position.X <= Control.Center.X + BoxSize.X * 0.5f &&
    Position.Y >= Control.Center.Y - BoxSize.Y * 0.5f &&
    Position.Y >= Control.Center.Y - BoxSize.Y * 0.5f;
}
```

Предупреждение PVS-Studio: V501 There are identical sub-expressions 'Position.Y >= Control.Center.Y — BoxSize.Y * 0.5f' to the left and to the right of the '&&' operator. svirtualjoystick.cpp 97

Примеры ошибок в коде движка Unreal Engine

```
static bool PositionIsInside(....)
{
  return
    Position.X >= Control.Center.X - BoxSize.X * 0.5f &&
    Position.X <= Control.Center.X + BoxSize.X * 0.5f &&
    Position.Y >= Control.Center.Y - BoxSize.Y * 0.5f &&
    Position.Y >= Control.Center.Y - BoxSize.Y * 0.5f;
    Position.Y <= Control.Center.Y + BoxSize.Y * 0.5f;
}
```

Предупреждение PVS-Studio: V501 There are identical sub-expressions 'Position.Y >= Control.Center.Y — BoxSize.Y * 0.5f' to the left and to the right of the '&&' operator. svirtualjoystick.cpp 97

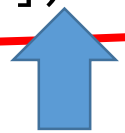
```
enum ECubeFace;
ECubeFace CubeFace;
friend FArchive& operator<<(FArchive& Ar, FResolveParams& ResolveParams)
  {
    ....
    if(Ar.IsLoading())
    {
      ResolveParams.CubeFace = (ECubeFace)ResolveParams.CubeFace;
    }
    ....
  }
```

```
enum ECubeFace;
ECubeFace CubeFace;
friend FArchive& operator<<(FArchive& Ar, FResolveParams&
ResolveParams)
  {
    ....
    if(Ar.IsLoading())
    {
      ResolveParams.CubeFace = (ECubeFace)ResolveParams.CubeFace;
    }
    ....
  }
```

Предупреждение PVS-Studio: V570 The 'ResolveParams.CubeFace' variable is assigned to itself. rhi.h 1279

```
bool VertInfluencedByActiveBone(
  FParticleEmitterInstance* Owner, USkeletalMeshComponent* InSkelMeshComponent,
  int32 InVertexIndex, int32* OutBoneIndex = NULL);
void UParticleModuleLocationSkelVertSurface::Spawn(....)
{
  int32 BoneIndex1, BoneIndex2, BoneIndex3;
  BoneIndex1 = BoneIndex2 = BoneIndex3 = INDEX_NONE;
  if(!VertInfluencedByActiveBone(
        Owner, SourceComponent, VertIndex[0], &BoneIndex1) &&
      !VertInfluencedByActiveBone(
        Owner, SourceComponent, VertIndex[1], &BoneIndex2) &&
      !VertInfluencedByActiveBone(
        Owner, SourceComponent, VertIndex[2]) &BoneIndex3)
   {
   ....
}
```

```
bool VertInfluencedByActiveBone(
  FParticleEmitterInstance* Owner, USkeletalMeshComponent* InSkelMeshComponent,
  int32 InVertexIndex, int32* OutBoneIndex = NULL);
void UParticleModuleLocationSkelVertSurface::Spawn(....)
{
  int32 BoneIndex1, BoneIndex2, BoneIndex3;
  BoneIndex1 = BoneIndex2 = BoneIndex3 = INDEX_NONE;
  if(!VertInfluencedByActiveBone(
        Owner, SourceComponent, VertIndex[0], &BoneIndex1) &&
      !VertInfluencedByActiveBone(
        Owner, SourceComponent, VertIndex[1], &BoneIndex2) &&
      !VertInfluencedByActiveBone(
        Owner, SourceComponent, VertIndex[2]) &BoneIndex3)
  {
  ....
}
```



Сообщение анализатора к хитрой ошибке

Предупреждение PVS-Studio: V564 The '&' operator is applied to bool type value. You've probably forgotten to include parentheses or intended to use the '&&' operator. particlemodules_location.cpp 2120

```
void GetRenderData(....)
{
  ....
  FT_Bitmap* Bitmap = nullptr;
  if( Slot->bitmap.pixel_mode == FT_PIXEL_MODE_MONO )
  {
    FT_Bitmap NewBitmap;
    ....
    Bitmap = &NewBitmap;
  }
  ....
  OutRenderData.RawPixels.AddUninitialized(
    Bitmap->rows * Bitmap->width );
  ....
}
```

```
void GetRenderData(....)
{
  ....
  FT_Bitmap* Bitmap = nullptr;
  if( Slot->bitmap.pixel_mode == FT_PIXEL_MODE_MONO )
  {
    FT_Bitmap NewBitmap;
    ....
    Bitmap = &NewBitmap;
  }
  ....
  OutRenderData.RawPixels.AddUninitialized(
    Bitmap->rows * Bitmap->width );
  ....
}
```

PVS-Studio: V506 Pointer to local variable 'NewBitmap' is stored outside the scope of this variable. Such a pointer will become invalid. fontcache.cpp 466

```
void UGameplayStatics::DeactivateReverbEffect(....)
{
  if (GEngine || !GEngine->UseSound())
  {
    return;
  }

  UWorld* ThisWorld = GEngine->GetWorldFromContextObject(....);
  ....
}
```

```
void UGameplayStatics::DeactivateReverbEffect(....)
{
  if (GEngine || !GEngine->UseSound())
  {
    return;
  }

  UWorld* ThisWorld = GEngine->GetWorldFromContextObject(....);
  ....
}
```

PVS-Studio: V522 Dereferencing of the null pointer 'GEngine' might take place. Check the logical condition. gameplaystatics.cpp 988

```
void UGameplayStatics::DeactivateReverbEffect(....)
{
  if (GEngine || !GEngine->UseSound())
  if (GEngine == nullptr || !GEngine->UseSound())
  {
    return;
  }

  UWorld* ThisWorld = GEngine->GetWorldFromContextObject(....);
  ....
}
```

PVS-Studio: V522 Dereferencing of the null pointer 'GEngine' might take place. Check the logical condition. gameplaystatics.cpp 988

```
template< typename DefinitionType >
FORCENOINLINE void Set(....)
{
  ....
  if ( DefinitionPtr == NULL )
  {
    WidgetStyleValues.Add( PropertyName,
      MakeShareable( new DefinitionType( InStyleDefintion ) ) );
  }
  else
  {
    WidgetStyleValues.Add( PropertyName,
      MakeShareable( new DefinitionType( InStyleDefintion ) ) );
  }
}
```

```
template< typename DefinitionType >
FORCENOINLINE void Set(....)
{
  ....
  if ( DefinitionPtr == NULL )
  {
    WidgetStyleValues.Add( PropertyName,
      MakeShareable( new DefinitionType( InStyleDefintion ) ) );
  }
  else
  {
    WidgetStyleValues.Add( PropertyName,
      MakeShareable( new DefinitionType( InStyleDefintion ) ) );
  }
}
```

PVS-Studio: V523 The 'then' statement is equivalent to the 'else' statement. paths.cpp 703

Заключение

- Статический анализ UE проектов это не так сложно
- Статический анализ должен сочетаться с другими методами тестирования

Спасибо за внимание