

인하우스 툴 포트폴리오

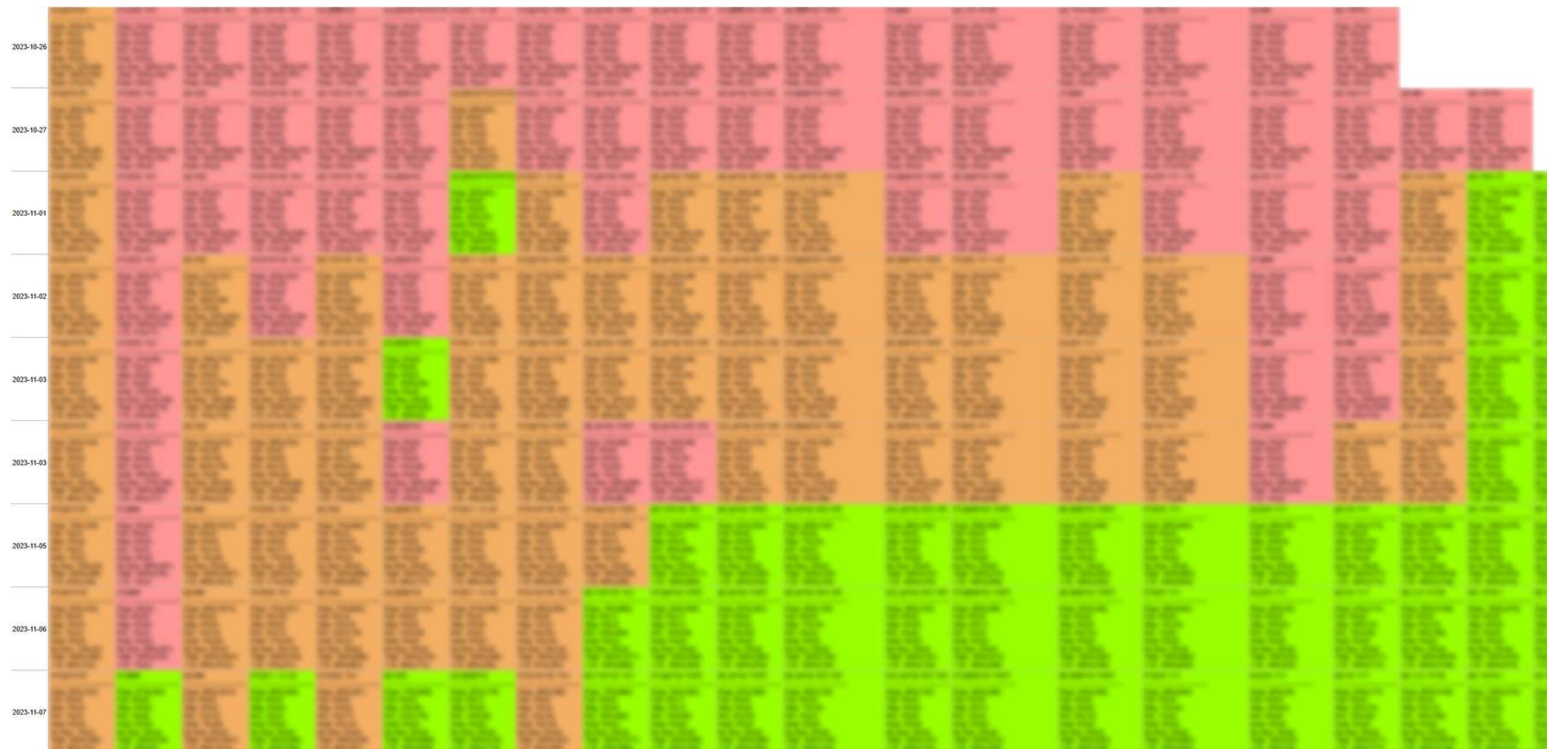
김영성

목차

- 1 표지
- 2 목차
- 3 업무진행 상황 실시간 누적 시각화
- 4 연차 일 때 진행상황 확인을 위해 만든 봇
- 5 멘티스 링크 간단한 가독성 갱신
- 6 회고를 위한 버전 진행도 & 이슈 리포트 자동화
- 7 영상 업로드를 위한 인코딩 프로그램
- 8 Thanks You

히스토리 관리

- 구글 시트의 경우 많은 데이터의 변경점을 누적하는곳이 없어 진행도 파악에 시간을 사용해야했습니다.
- 실시간으로 변경되는 커버리지 데이터를 한곳에 집계하여 일정 시간 마다, 히스토리화 시켰습니다.



```

1 function recordDailyValues() {
2   var ss = SpreadsheetApp.getActiveSpreadsheet();
3   var sheets = ss.getSheets();
4   var keywords = ["[TC]", "[CL]", "[D]"];
5   var matchedSheetsData = [];
6   var labels = ["Pass", "Fail", "Skip", "N/A", "Hold", "No Run", "Total"];
7
8   for (var i = 0; i < sheets.length; i++) {
9     var sheet = sheets[i];
10    var sheetName = sheet.getName();
11    for (var j = 0; j < keywords.length; j++) {
12      if (sheetName.indexOf(keywords[j]) !== -1) {
13        var values = sheet.getRange("C3:C9").getValues();
14        var total = values[6][0];
15        var formattedValues = labels.map((label, index) => {
16          var percentage = Math.round((values[index][0] / total) * 100);
17          return label + ": " + percentage + "%(" + values[index][0] + ")";
18        });
19
20        var sumWithoutNoRun = 0;
21        for (var k = 0; k < labels.length; k++) {
22          if (labels[k] === "Pass" || labels[k] === "N/A" || labels[k] === "Skip") {
23            sumWithoutNoRun += values[k][0];
24          }
25        }
26
27        var percentageWithoutNoRun = Math.round((sumWithoutNoRun / total) * 100);
28        formattedValues.push("수행: " + percentageWithoutNoRun + "%(" + values[labels.indexOf("Pass")][0] + ")");
29        matchedSheetsData.push({
30          sheetName: sheetName + "\n-----",
31          values: formattedValues,
32          executionRate: percentageWithoutNoRun
33        });
34        break;
35      }
36    }
37  }

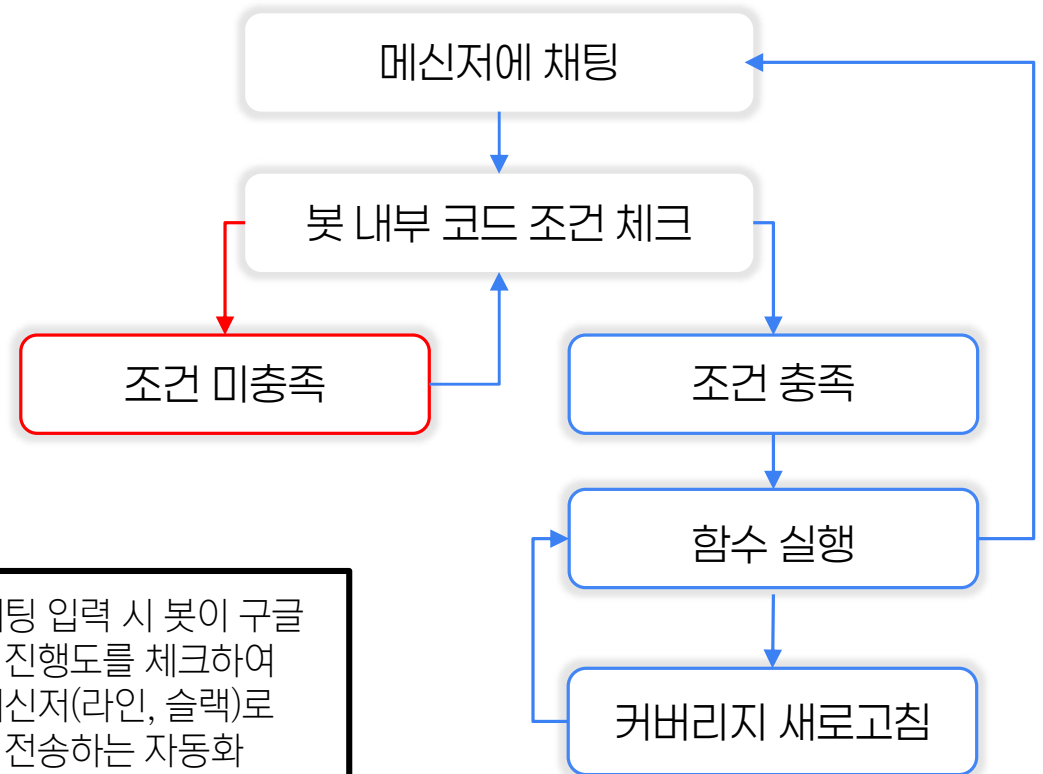
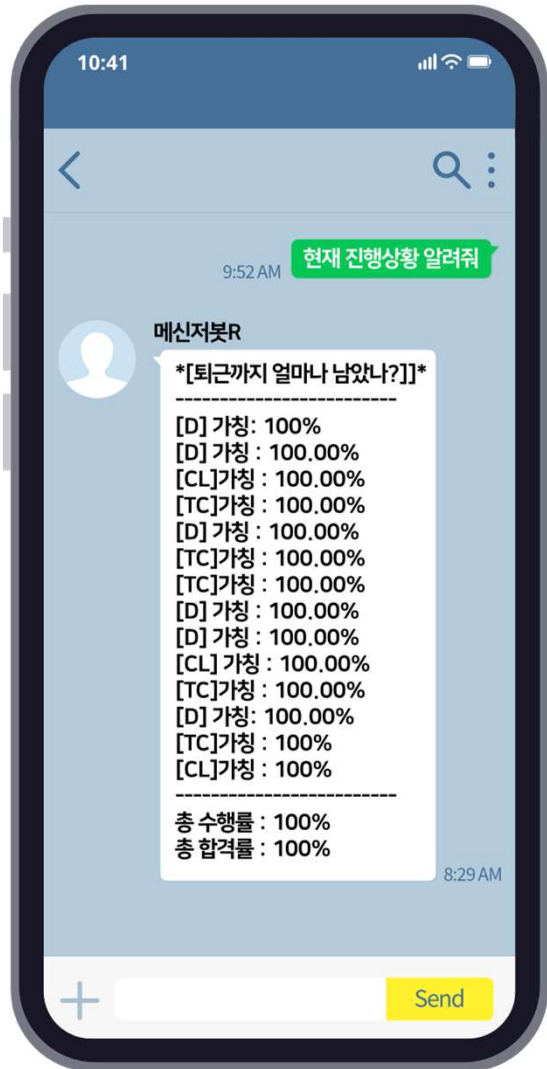
```

```

38
39  var recordSheet = ss.getSheetByName("일차 별 대시보드");
40  if (!recordSheet) {
41    recordSheet = ss.insertSheet("일차 별 대시보드");
42    recordSheet.appendRow(["Date"].concat(keywords));
43  }
44
45  var today = new Date();
46  var formattedDate = Utilities.formatDate(today, ss.getSpreadsheetTimeZone(), "yyyy-MM-dd");
47  var newRow = [formattedDate];
48
49  for (var i = 0; i < matchedSheetsData.length; i++) {
50    var currentData = matchedSheetsData[i].sheetName + "\n" + matchedSheetsData[i].values.join("\n");
51    newRow.push(currentData);
52  }
53
54  recordSheet.appendRow(newRow);
55  var lastRowIndex = recordSheet.getLastRow();
56
57  for (var i = 0; i < matchedSheetsData.length; i++) {
58    var executionRate = matchedSheetsData[i].executionRate;
59    if (executionRate >= 100) {
60      recordSheet.getRange(lastRowIndex, i + 2).setBackground("#99FF00");
61    } else if (executionRate >= 50 && executionRate < 100) {
62      recordSheet.getRange(lastRowIndex, i + 2).setBackground("#F5AF64");
63    } else if (executionRate <= 49 && executionRate >= 0) {
64      recordSheet.getRange(lastRowIndex, i + 2).setBackground("#FF9696");
65    }
66  }
67 }
68

```

연차 일 때 진행상황 확인을 위해 만든 봇 04



특정 채팅 입력 시 봇이 구글 시트의 진행도를 체크하여 업무 메신저(라인, 슬랙)로 알림을 전송하는 자동화 시스템을 구축했습니다.

```

1  var count = 4;
2
3  function SendCoverageNotify() {
4      token = [" "];
5      payloadText = "message=" + "\n\n" +
6          "[퇴근까지 얼마나 남았나?]" + "\n" +
7          "-----" + "\n" +
8          MakeMessage();
9      sendtoslack(payloadText);
10 }
11
12 function sendtoslack(payloadText) {
13     var timestamp = new Date();
14     var url = " ";
15     var payload = {
16         "channel": "",
17         "attachments": [{
18             "text": payloadText,
19             "mrkdwn_in": ["text"]
20         }],
21     };
22
23     var options = {
24         "method": "post",
25         "contentType": "application/json",
26         "payload": JSON.stringify(payload)
27     };
28
29     return UrlFetchApp.fetch(url, options);
30 }

```

```

32 function MakeMessage() {
33     var summarySheet = spreadsheet.getSheetByName("테스트 요약");
34     var pass = summarySheet.getRange('079');
35     var total = summarySheet.getRange('N79');
36     var message = "";
37     var coveragePercentages = summarySheet.getRange(57, 14, 19).getValues();
38
39     for (var i = 0; i < sheetNames.length - 4; i++) {
40         var percentage = Number(coveragePercentages[i]);
41         var addMessage = "";
42         if (coveragePercentages[i] == "") {
43             count++;
44         }
45         if (coveragePercentages[i] != "") {
46             addMessage = sheetNames[i + count + 2] + " : " + (percentage * 100).toFixed(2) + "%\n";
47         }
48         message += addMessage;
49     }
50
51     message += "-----" + "\n" +
52         "총 수행률 : " + Math.round(total.getValue() * 100) + "%\n" +
53         "총 합격률 : " + Math.round(pass.getValue() * 100) + "%";
54     return message;
55 }
56
57 function Fetch(payloadText, token) {
58     var options = {
59         "method": "post",
60         "payload": payloadText,
61         "headers": {
62             "Authorization": "Bearer " + token
63         }
64     };
65     UrlFetchApp.fetch("https://notify-api.line.me/api/notify", options);
66 }
67

```

```

1  var spreadsheet = SpreadsheetApp.getActiveSpreadsheet();
2  var sheet = SpreadsheetApp.getActiveSheet();
3  var sheets = spreadsheet.getSheets();
4  var sheetNames = GetSheetNames();
5  var bold = SpreadsheetApp.newTextStyle().setBold(true).build();
6  try {
7    var ui = SpreadsheetApp.getUi();
8  } catch (error) {}
9
10 var cell = sheet.getActiveCell();
11 var str = cell.getValue();
12
13 function onEdit111111Function(e) {}
14
15 function BoldBrackets() {
16   var richTextValue = cell.getRichTextValue();
17   var originalTextStyle = richTextValue.getRuns().map(e => ({
18     start: e.getStartIndex(),
19     end: e.getEndIndex(),
20     style: e.getTextStyle()
21   }));
22
23   var richTextValueBuilder = SpreadsheetApp.newRichTextValue().setText(str);
24   originalTextStyle.forEach(e => {
25     richTextValueBuilder.setTextStyle(e.start, e.end, e.style)
26   });
27
28   var startPoint = 0;
29   var endPoint = 0;
30   var tempPoint = 0;
31
32   while (true) {
33     startPoint = str.indexOf("[", tempPoint);
34     if (startPoint === -1) break;
35
36     endPoint = str.indexOf("]", tempPoint);
37     tempPoint = endPoint + 1;
38
39     richTextValueBuilder.setTextStyle(startPoint, endPoint + 1, bold);
40   }
41
42   cell.setRichTextValue(richTextValueBuilder.build());
43 }
44

```

```

45 function SheetLinker() {
46   var currentSheetName = sheet.getSheetName();
47
48   for (var i = 0; i < sheetNames.length; i++) {
49     var targetName = sheetNames[i][0];
50     if (str.includes(targetName) && targetName !== currentSheetName) {
51       var refSheetId = spreadsheet.getSheetByName(targetName).getSheetId().toString();
52       var link = `${spreadsheet.getUrl()}#gid=${refSheetId}`;
53       var formula = `=HYPERLINK("${link}", "${str}")`;
54       cell.setFormula(formula).setFontColor('blue');
55       break;
56     }
57   }
58 }
59
60 function LastEditTime() {
61   var nowDate = Utilities.formatDate(new Date(), 'Korea/Seoul', "YYYY-MM-dd");
62
63   if (!sheet.getRange('B2').getValue()) {
64     sheet.getRange('B1').setValue("First Write : " + nowDate);
65   }
66
67   sheet.getRange('B2').setValue("Last Update : " + nowDate);
68 }
69
70 function replaceAll(str, searchStr, replaceStr) {
71   return str.split(searchStr).join(replaceStr);
72 }
73
74 function compareSheetName(sheetName) {
75   return spreadsheet.getSheetByName(sheetName) !== null;
76 }
77
78 function GetSheetNames() {
79   var out = [];
80   var sheets = spreadsheet.getSheets();
81   for (var i = 0; i < sheets.length; i++) {
82     out.push([sheets[i].getName()]);
83   }
84   sheetNames = out;
85   return out;
86 }

```

[]브라켓 안 내용 볼드

시트 이름에 포함되면 링크

수정 일자 기록

멘티스 링크 입력 시 굵은 글씨(Bold)로 가독성을 높이고, 하이퍼링크로 전환하여 접근성을 향상 시켰습니다.

또한 업데이트 날짜를 갱신시켜,

시트 진입 시 확인 할 수 있게, 편의성을 개선했습니다.

일차·시트별 업무 현황을 한눈에
확인하고, 실시간 커버리지
히스토리를 누적해 시각화합니다.

Component	합계 건수			Status							수행률	합계율
	Total	마수행	수행	Pass	Fail	Skip	N/A	Hold	No Run			
[T]가칭	164	4	160	150	0	4	2	0	0	100,00%	97,53%	
[T]가칭	210	0	210	204	0	0	6	0	0	100,00%	100,00%	
[M]가칭	1647	0	1647	1633	0	0	14	0	0	100,00%	100,00%	
[T]가칭	318	0	318	301	0	0	17	0	0	100,00%	100,00%	
[M]가칭	808	20	788	609	0	20	179	0	0	100,00%	96,82%	
[C]가칭	210	0	210	190	0	0	30	0	0	100,00%	100,00%	
[T]가칭	332	4	328	312	0	4	16	0	0	100,00%	98,73%	
[M]가칭	1377	4	1373	960	0	4	405	0	0	100,00%	99,59%	
[T]가칭	265	0	265	260	0	0	5	0	0	100,00%	100,00%	
[T]가칭	358	1	357	301	0	1	56	0	0	100,00%	99,67%	
[M]가칭	390	0	390	360	0	0	30	0	0	100,00%	100,00%	
[M]가칭	135	10	125	120	0	10	5	0	0	100,00%	92,31%	
[C]가칭	141	0	141	135	0	0	6	0	0	100,00%	100,00%	
[T]가칭	214	1	213	209	0	1	4	0	0	100,00%	99,52%	
[M]가칭	263	0	263	240	0	0	15	0	0	100,00%	100,00%	
[T]가칭	962	10	952	944	0	10	0	0	0	100,00%	98,95%	
[C]가칭	25	5	20	19	0	0	1	0	0	100,00%	96,00%	
[M]가칭	271	70	201	173	0	70	28	0	0	100,00%	71,19%	
[M]가칭	3919	13	3906	3145	0	0	0	0	0	100,00%	99,59%	
[M]가칭	476	0	476	476	0	0	0	0	0	100,00%	100,00%	
[M]가칭	1423	300	1043	1039	4	300	0	0	0	100,00%	73,01%	
	0	0	0									
	0	0	0									
	0	0	0									
	0	0	0									
	0	0	0									
	0	0	0									
Total	13908	522	13386	11796	4	522	1586	0	0	100%	96%	

Dash Board

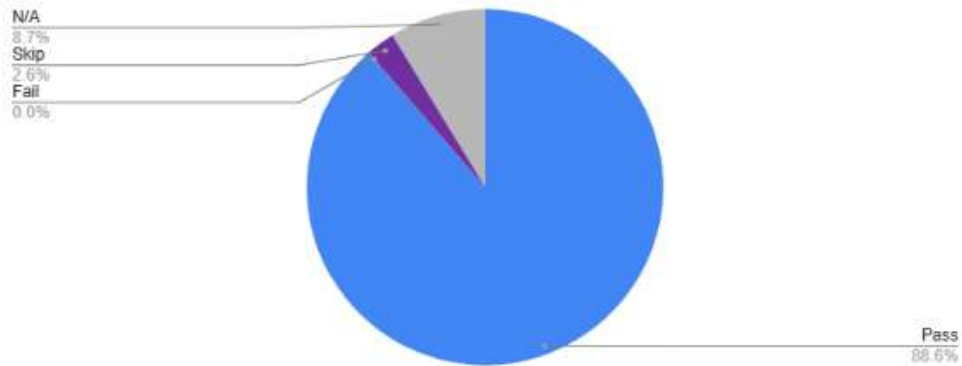
대시보드 새로고침

데이터 갱신

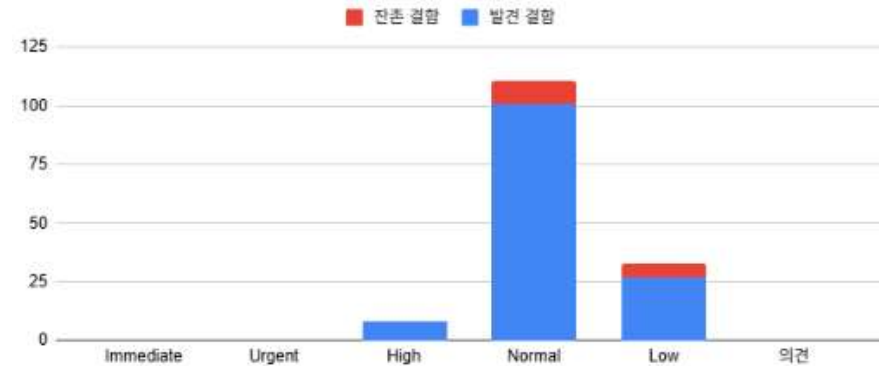
조건부 서식
활용

시각화

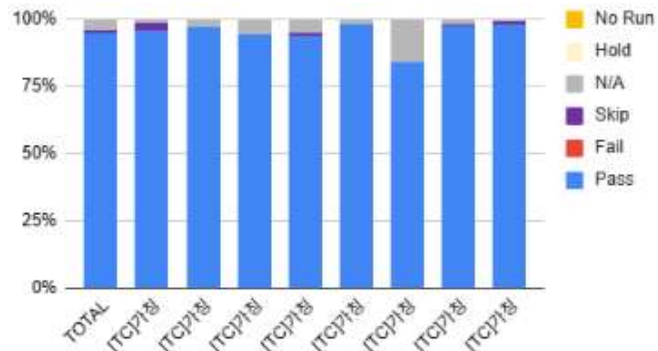
Result Graph



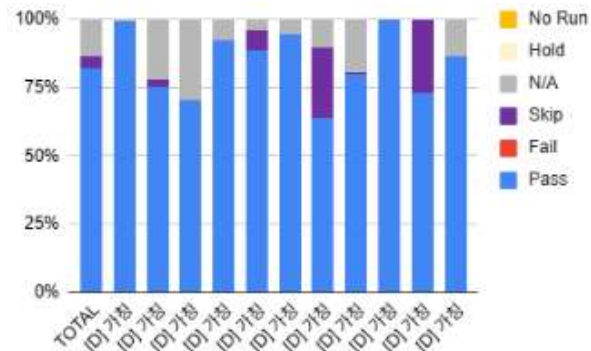
Defects Graph



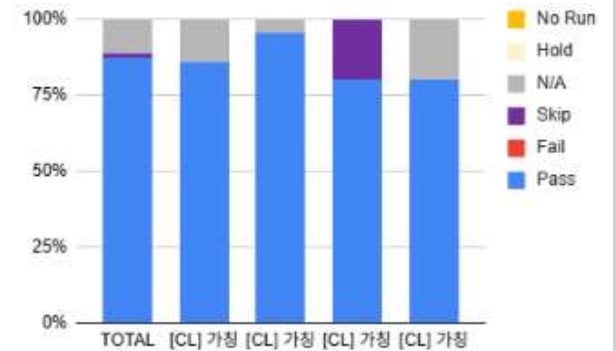
TC_Component Result Graph



Data_Component Result Graph



CL_Component Result Graph



Component	항목 건수			Status						수행률	합격률
	Total	미수행	수행	Pass	Fail	Skip	N/A	Hold	No Run		
[TC]가칭	164	4	160	158	0	4	2	0	0	100,00%	97,53%
[TC]가칭	210	0	210	204	0	0	6	0	0	100,00%	100,00%
[D]가칭	1647	0	1647	1633	0	0	14	0	0	100,00%	100,00%
[TC]가칭	318	0	318	301	0	0	17	0	0	100,00%	100,00%
[D]가칭	808	20	788	609	0	20	179	0	0	100,00%	96,82%
[CL]가칭	210	0	210	180	0	0	30	0	0	100,00%	100,00%
[TC]가칭	332	4	328	312	0	4	16	0	0	100,00%	98,73%
[D]가칭	1377	4	1373	968	0	4	405	0	0	100,00%	99,59%
[TC]가칭	265	0	265	260	0	0	5	0	0	100,00%	100,00%
[TC]가칭	358	1	357	301	0	1	56	0	0	100,00%	99,67%
[D]가칭	390	0	390	360	0	0	30	0	0	100,00%	100,00%
[D]가칭	135	10	125	120	0	10	5	0	0	100,00%	92,31%
[CL]가칭	141	0	141	135	0	0	6	0	0	100,00%	100,00%
[TC]가칭	214	1	213	209	0	1	4	0	0	100,00%	99,52%
[D]가칭	263	0	263	248	0	0	15	0	0	100,00%	100,00%
[TC]가칭	962	10	952	944	0	10	8	0	0	100,00%	98,95%
[CL]가칭	25	5	20	20	0	5	0	0	0	100,00%	80,00%
[D]가칭	271	70	201	173	0	70	28	0	0	100,00%	71,19%
[D]가칭	3919	13	3906	3146	0	13	760	0	0	100,00%	99,59%
[D]가칭	476	0	476	476	0	0	0	0	0	100,00%	100,00%
[D]가칭	1423	380	1043	1039	4	380	0	0	0	100,00%	73,01%
	0	0	0								
	0	0	0								
	0	0	0								
	0	0	0								
	0	0	0								
Total	13908	522	13386	11796	4	522	1586	0	0	100%	96%

커버리지 데이터를 신속하고 정확하게 파악할 수 있었고, 회고 진행 시 필요한 데이터를 직관적으로 확인할 수 있어 편리했고 무엇보다 데이터를 정리하는데 시간이 소요되지 않는 점이 좋았습니다.

```

1  var dashBoardSheet = spreadsheet.getSheetByName("대시보드");
2  var countTC = 0;
3  var countData = 0;
4  var countCL = 0;
5
6  function ResetDate() {
7    for (i = 0; i < sheets.length; i++) {
8      sheets[i].getRange('B1').setValue("");
9      sheets[i].getRange('B2').setValue("");
10   }
11 }
12
13 function RefreshCoverage() {
14   var sheetNames = GetSheetNames();
15   dashBoardSheet.getRange("대시보드!G16:V22").clearContent();
16   dashBoardSheet.getRange("대시보드!G26:V32").clearContent();
17   dashBoardSheet.getRange("대시보드!G36:V42").clearContent();
18   dashBoardSheet.getRange("테스트 요약!C66:D90").clearContent();
19   countTC = 0;
20   countData = 0;
21   countCL = 0;
22
23   for (i = 0; i < sheetNames.length; i++) {
24     WhereIsEmptyCellInDashBoardSheetNames(sheetNames[i][0]);
25   }
26
27   RefreshSummarySheet();
28   spreadsheet.toast("갱신 완료");
29 }
30
31 function WhereIsEmptyCellInDashBoardSheetNames(sheetName) {
32   var kindOfSheet = WhatKindOfSheet(sheetName);
33   switch (kindOfSheet) {
34     case "TC":
35       dashBoardSheet.getRange(16, countTC + 7).setValue(sheetName);
36       RewriteCoverage(sheetName, 16, countTC + 7);
37       countTC++;
38       break;

```

```

39     case "[D]":
40       dashBoardSheet.getRange(26, countData + 7).setValue(sheetName);
41       RewriteCoverage(sheetName, 26, countData + 7);
42       countData++;
43       break;
44     case "[CL]":
45       dashBoardSheet.getRange(36, countCL + 7).setValue(sheetName);
46       RewriteCoverage(sheetName, 36, countCL + 7);
47       countCL++;
48       break;
49   }
50 }
51
52 function RewriteCoverage(sheetName, rowNum, cloumnNum) {
53   var tempSheet = spreadsheet.getSheetByName(sheetName);
54   for (j = 1; j <= 6; j++) {
55     var value = tempSheet.getRange(j + 2, 3).getValue();
56     dashBoardSheet.getRange(j + rowNum, cloumnNum).setValue(value);
57   }
58 }
59
60 function RefreshSummarySheet() {
61   var summarySheet = spreadsheet.getSheetByName("테스트 요약");
62   for (k = 0; k <= sheetNames.length - 8; k++) {
63     summarySheet.getRange(66 + k, 3).setValue(sheetNames[k + 4]);
64   }
65 }
66
67 function WhatKindOfSheet(sheetName) {
68   if (sheetName.indexOf("TC") > -1) return "TC";
69   else if (sheetName.indexOf("[D]") > -1) return "[D]";
70   else if (sheetName.indexOf("[CL]") > -1) return "[CL]";
71   else return "None";
72 }
73

```

파일 용량 | 68.4MB

**인코딩 약 18s
45%**

BTS를 영상 업로드 용량 제한으로 인하여,
Python과 FFmpeg를 이용하여 영상
인코딩 프로그램 제작
프로그램의 간단한 사용을 위해
**파일위치의 폴더에 있는 영상들만 인코딩
하는 방식을 사용했습니다.**

파일 용량 | 37.3MB

실행화면

```

--enable-libtheora --enable-libvpx --enable-libwebp --enable-lv2 --enable-libvpl --enable-libopenh264 --enable-libopenh264 --enable-libopenh264
--enable-libopenjpeg --enable-libopenmpt --enable-librav1e --enable-librubberband --enable-sdl2 --enable-libsoxr --enable-libstb --enable-libsvtav1 --enable-libtesseract
--enable-libva3d --enable-libvdpau --enable-libvorbis --enable-libvidstab --enable-vulkan --enable-libshaderc --enable-liblilabcebo --enable-libx264 --enable-libx265 --enable-libxavs2
--enable-libxvid --enable-libzimg --enable-libzmb --extra-cflags=-DLIBWOLAME_STATIC --extra-cxxflags= --extra-ldflags=-pthread --extra-ldexeflags= --extra-libs=-lgomp --extra-y
version=20230920
libavutil 58. 25.100 / 58. 25.100
libavcodec 60. 26.100 / 60. 26.100
libavformat 60. 13.100 / 60. 13.100
libavdevice 60. 2.101 / 60. 2.101
libavfilter 9. 11.100 / 9. 11.100
libbswscale 7. 3.100 / 7. 3.100
libswresample 4. 11.100 / 4. 11.100
libpostproc 57. 2.100 / 57. 2.100
Input #0: mov,mp4,m4a,3gp,3g2,mj2, from 'C:/Users/KimYeongSung/Documents/카카오톡 받은 파일/VideoCompressor/1718887518_sample_1280x720_surfing_with_audio.mp4':
Metadata:
  major_brand      : isom
  minor_version    : 512
  compatible_brands: isomiso2avc1mp41
  encoder          : Lavf58.45.100
Duration: 00:03:03.13, start: 0.000000, bitrate: 3134 kb/s
Stream #0:0[0x1](und): Video: h264 (High) (avc1 / 0x31637661), yuv420p(progressive), 1280x720 [SAR 1:1 DAR 16:9], 3001 kb/s, 23.98 fps, 23.98 tbr, 24k tbn (default)
Metadata:
  handler_name     : VideoHandler
  vendor_id        : [01][00][00]
Stream #0:1[0x2](eng): Audio: aac (LC) (mp4a / 0x6134706D), 48000 Hz, stereo, fltp, 128 kb/s (default)
Metadata:
  handler_name     : Stereo
  vendor_id        : [01][00][00][00]
File 'C:/Users/KimYeongSung/Documents/카카오톡 받은 파일/VideoCompressor/1718887518_sample_1280x720_surfing_with_audio_encoding.mp4' already exists. Overwrite? [y/N]

```

Thanks You